



[Software Verification] Static analysis & System testing

Team 4

201411259 교수창

201411314 전소영

201412005 이세라

201511304 하지윤



Index

1. Static analysis

- Sonarqube
- CheckStyle
- PMD
- FindBugs / SpotBugs
- Error Prone

2. System testing

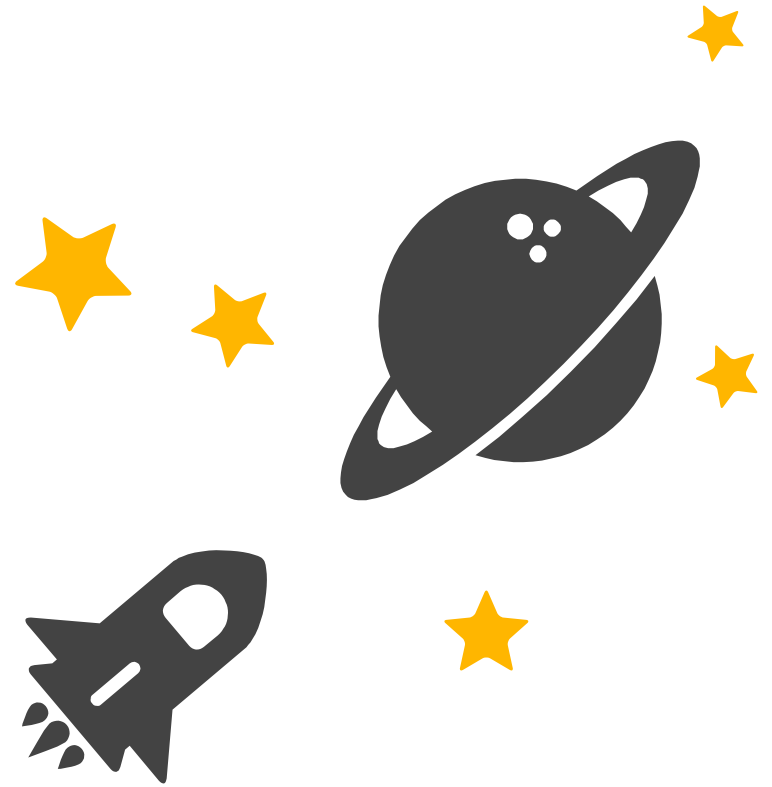
- TestLink

3. CTIP Environment

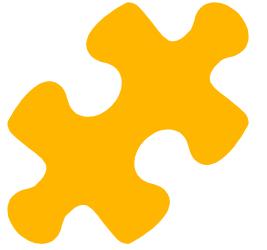
4. Q & A

Static analysis

정적 분석은 프로그램을 실행하지 않고, 코드 레벨에서 분석하는 방법



Static analysis



- Code review
- Coding convention (코드 작성 규칙)
- Byte code 기반 버그 패턴 방지
- Package 의존성 측정
- 사용하지 않는 변수, 불필요한 객체 등 검사

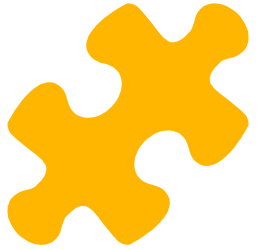
Sonarqube

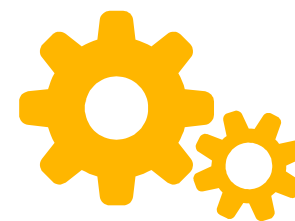
www.sonarqube.org

- 프로그램 소스의 정적 분석 도구
- PMD, CheckStyle, FindBugs 등을 플러그인으로 쉽게 연동 가능
- Jenkins와의 연동으로 지속적인 정적분석 작업이 가능
- 신뢰성(버그) / 취약성(보안) / 유지보수의 규칙에 대해서 검사

Microsoft Azure cloud를 사용하여 환경 구축

sonarqube 



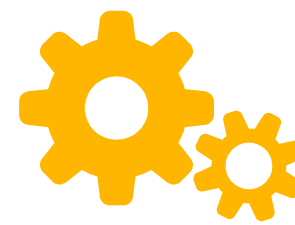


Sonarqube

1) Plugin 설치

Marketplace에서 PMD, FindBug, CheckStyle plugin 설치 가능

The screenshot displays the Sonarqube Administration interface, specifically the Marketplace section. The top navigation bar includes 'sonarqube', 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', and 'Administration'. A search bar is present on the right. The 'Administration' menu is expanded to show 'Configuration', 'Security', 'Projects', 'System', and 'Marketplace'. The 'Marketplace' section is titled 'Discover and install new features' and contains four cards for different editions: 'Community Edition' (marked as 'Installed'), 'Developer Edition', 'Enterprise Edition', and 'Data Center Edition'. Each card provides a brief description and an 'Upgrade' button. Below the editions, there are filters for 'All' and 'Updates Only', and a search input containing 'pmd'. A search result for 'PMD External Analysers' is shown, including a version number '2.6', a description of updates, and an 'Install' button.



Sonarqube

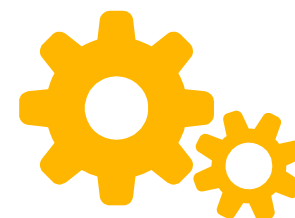
2) Quality Profile 설정

기존의 규칙도 사용 가능

The screenshot displays the Sonarqube 'Quality Profiles' management interface. A 'New Profile' modal dialog is open, allowing the user to create a new profile. The dialog includes the following fields and options:

- Name***: Text input field containing 'Coding convention'.
- Language***: Dropdown menu set to 'Java'.
- PMD**: Checkable option with a '찾아보기...' (Search) button and an 'Optional configuration file' label.
- FindBugs**: Checkable option with a '찾아보기...' (Search) button and an 'Optional configuration file' label.
- Checkstyle**: Checkable option with a '찾아보기...' (Search) button and an 'Optional configuration file' label.

At the bottom of the dialog are 'Create' and 'Cancel' buttons. The background interface shows the 'Quality Profiles' page with a search bar and a list of profiles for different languages: C# (1 profile), Flex (1 profile), and Java (6 profiles). A 'Recently Added Rules' sidebar is also visible on the right.



Sonarqube

3) 규칙 활성화

Google convention 규칙 활성화

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

Search for rules... Bulk Change Clear All Filters

Language

Search

Type

- Bug 0
- Vulnerability 0
- Code Smell 0

Tag

Repository

Default Severity

Status

Available Since

Template

Quality Profile

- Drupal PHP (Built-in)
- FindBugs Java (Built-in)
- FindBugs + FB-Contrib Java (Built-in)
- FindBugs Security Audit Java (Built-in)
- FindBugs Security JSP JSP (Built-in)
- FindBugs Security Minimal Java (Built-in)
- Google Convention Java **active** inactive

FindBugs + FB-Contrib Java (Built-in)

FindBugs Security Audit Java (Built-in)

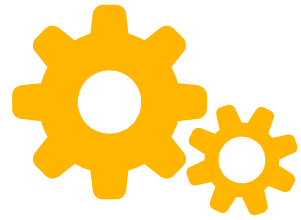
FindBugs Security JSP JSP (Built-in)

FindBugs Security Minimal Java (Built-in)

Google Convention Java **active** inactive

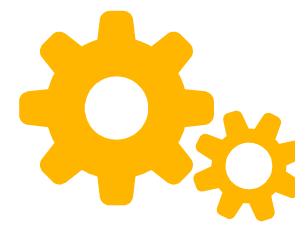
Sonarqube

4) Project의 quality profile 설정



The screenshot shows the Sonarqube interface for a project named 'ctip-year' in the 'master' branch. The 'Administration' tab is selected, and the 'Quality Profiles' section is active. The page instructs the user to choose a quality profile for each language. The current settings are as follows:

Language	Quality Profile
C#	Default: Sonar way
Flex	Default: Sonar way
JSP	Default: FindBugs Security JSP
Java	Google Convention

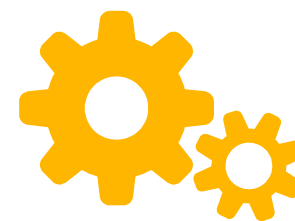


Sonarqube

5) Project dashboard

프로젝트에 존재하는 버그, 취약점, 코드 냄새에 대해서 간략하게 확인 가능

The screenshot shows the Sonarqube interface. At the top, there's a navigation bar with 'sonarqube' logo and tabs for 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', and 'Administration'. A search bar is on the right. Below the navigation, there are tabs for 'My Favorites' and 'All'. The main area shows the project 'ctip-year' with a 'Passed' status. Metrics are displayed: 0 Bugs (A), 0 Vulnerabilities (A), 5 Code Smells (A), 0.0% Coverage, and 0.0% Duplications. The last analysis was on April 13, 2018 at 4:05 AM. A '29 XS Java' badge is also visible. The bottom of the dashboard shows '1 of 1 shown'.



Sonarqube

6) Code Review 진행

ctip-yearh master

Overview Issues Measures **Code** Activity Administration

Search for files and sub-projects...

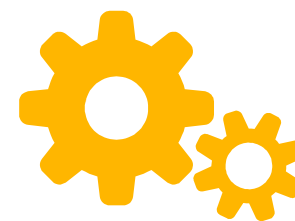
ctip-yearh > unit_test/src > JUnitTest.java

```
1 crypt... import org.junit.jupiter.api.AfterEach;
2 import org.junit.jupiter.api.BeforeEach;
3 import org.junit.jupiter.api.Test;
4
5 import java.time.temporal.JulianFields;
```

Remove this unused import 'java.time.temporal.JulianFields'. ...

Code Smell Minor Open Administrator 2min effort Comment

Administrator 사용하지 않으니까 지워야합니다.



Sonarqube

7) Code Smell 확인

The screenshot displays the Sonarqube interface for a project named 'ctip-year' on the 'master' branch. The current view is for the file 'unit_test/src/JUnit.java'. The interface shows a list of code smells on the left and a detailed view of the code with overlaid issue boxes on the right.

Code Smells List:

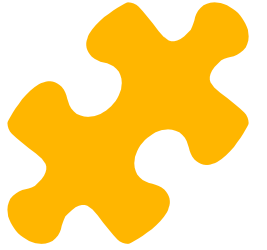
- unit_test/src/JUnit.java**
 - Move this file to a named package.** (Code Smell, Minor)
 - Rename this method name to match the regular expression `^[a-z][a-zA-Z0-9]*$`.** (Code Smell, Minor)
- unit_test/src/JUnitTest.java**
 - Move this file to a named package.** (Code Smell, Minor)
 - Remove this unused import `'java.time.temporal.JulianFields'`.**

Code Snippet and Issue Details:

```
1 crypt... public class JUnit {
2
3
4     public int junit_int(int num) {
```

Issue 1: Move this file to a named package. (Code Smell, Minor, Open, Not assigned, 10min effort, 2 hours ago)

Issue 2: Rename this method name to match the regular expression `^[a-z][a-zA-Z0-9]*$`. (Code Smell, Minor, Open, Not assigned, 5min effort, 2 hours ago)

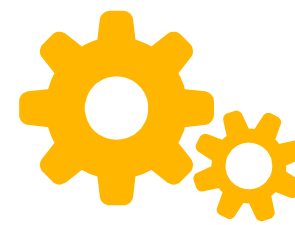


CheckStyle

checkstyle.sourceforge.net

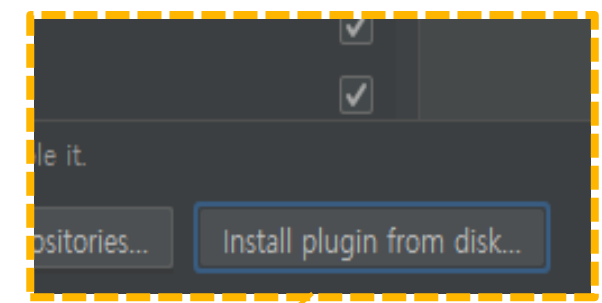
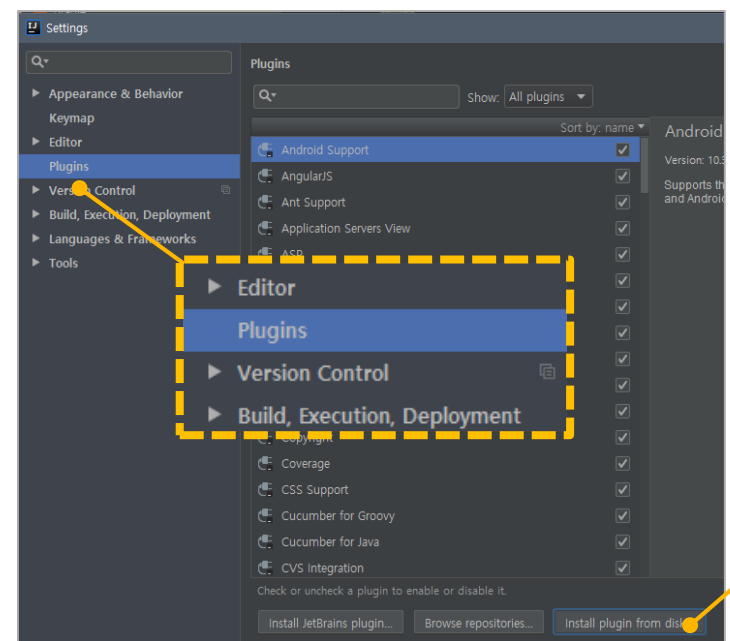
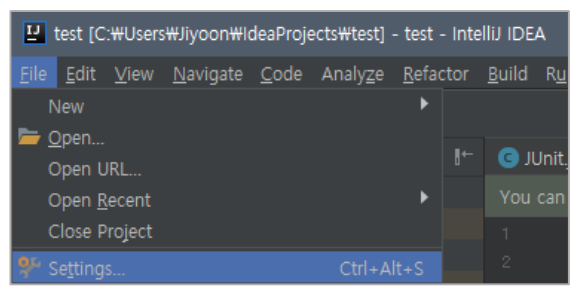
- 소스 코드에서 coding convention을 지켰는지 확인해주는 도구
- xml 파일 형식으로 정의된 규칙을 사용하여, 언제든지 다른 규칙을 적용하거나 수정할 수 있음
- Google이나 Twitter 등 기업에서 사용하는 공개된 convention을 사용할 수 있음

checkstyle

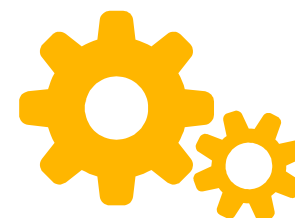


CheckStyle

1) IntelliJ IDEA plugin 설치 - 1



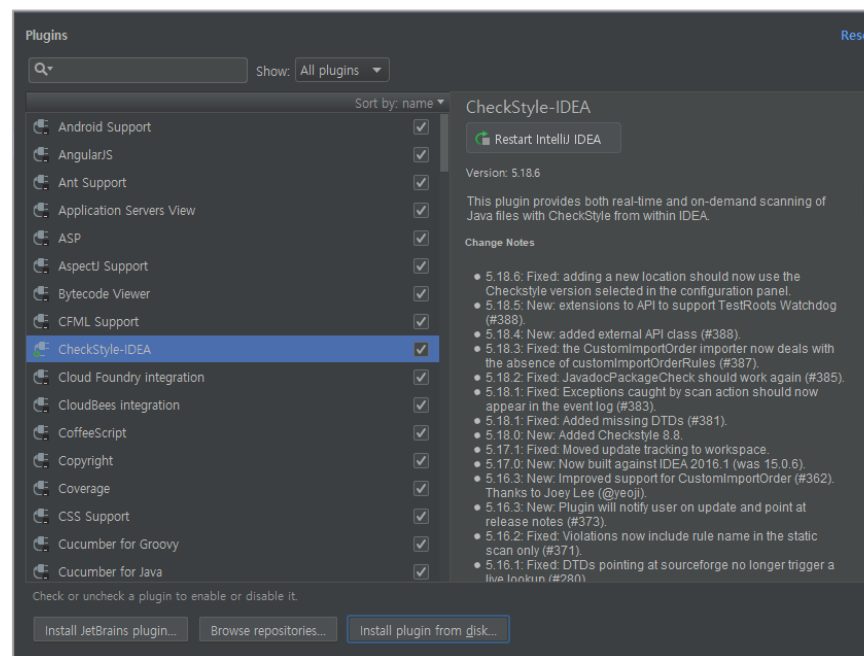
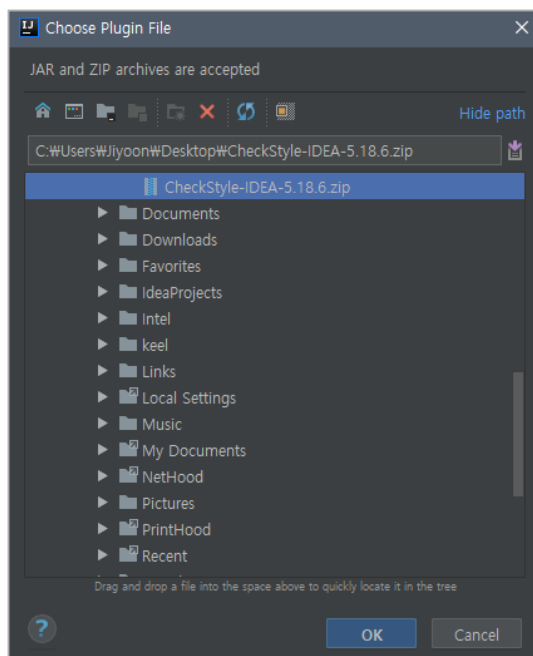
플러그인 설치: plugins.jetbrains.com/plugin/1065-checkstyle-idea

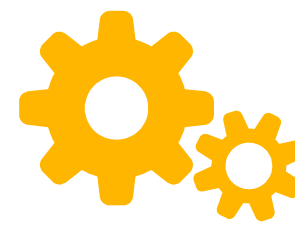


CheckStyle

2) IntelliJ IDEA plugin 설치 - 2

설치한 플러그인 압축파일 불러오기

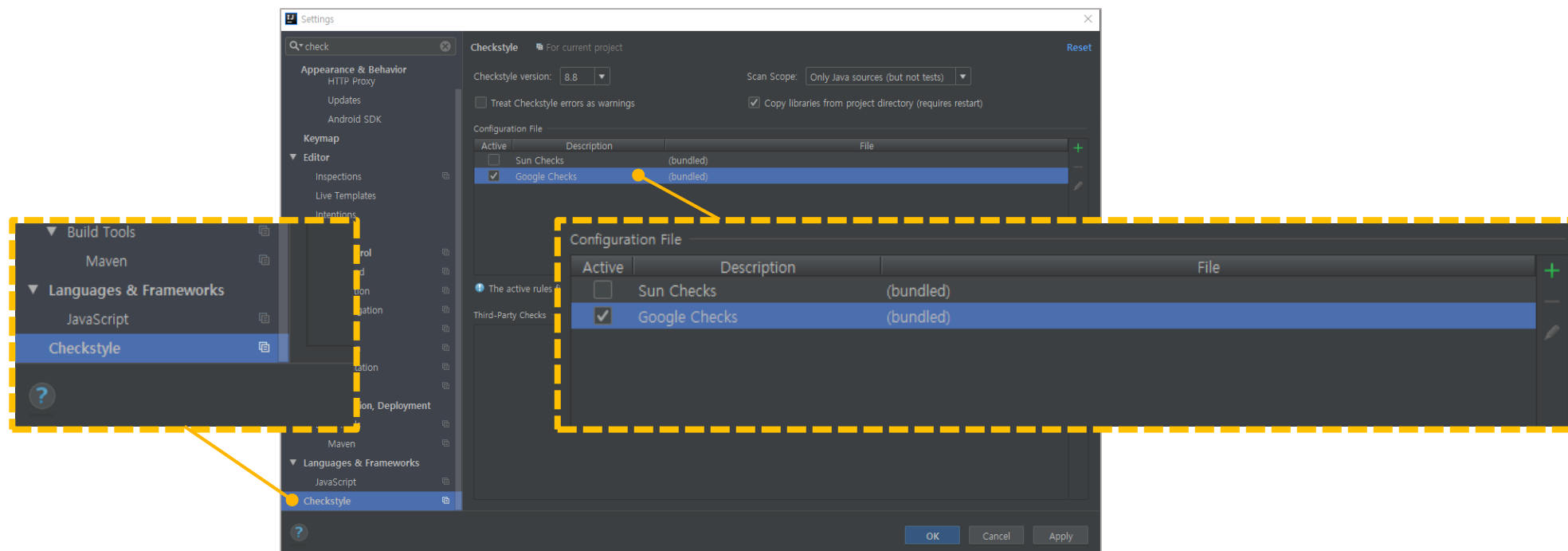




CheckStyle

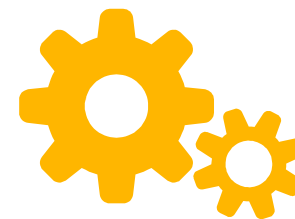
3) IntelliJ IDEA plugin 설정

기본으로 내장된 규칙으로는 Sun과 Google Checks가 존재



google java style guide: google.github.io/styleguide/javaguide.html

google checks: github.com/checkstyle/checkstyle/blob/master/src/main/resources/google_checks.xml



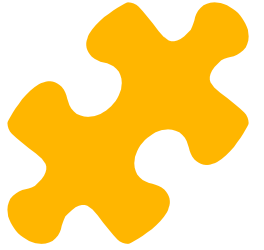
CheckStyle

4) Gradle script 작성

```
1 apply plugin: 'java'
2
3 checkstyle {
4     ignoreFailures = true
5     configFile = file('config/checkstyle/checkstyle.xml')
6     toolVersion = "8.8"
7     sourceSets {
8         main {
9             java {
10                srcDir 'unit_test/src'
11            }
12            resources {
13                srcDir 'unit_test/src'
14            }
15        }
16    }
17 }
18
19 checkstyleMain {
20     reports {
21         xml.enabled = true
22         html.enabled = false
23     }
24 }
25
26 sourceCompatibility = 1.8
27
28 repositories {
29     mavenCentral()
30 }
31
32 dependencies {
33     testCompile group: 'junit', name: 'junit', version: '4.12'
34     compile 'junit:junit:4.12'
35 }
```

build.gradle

- **configFile**
coding convention 규칙이 정의되어 있는 파일 설정
- **sourceSets**
분석할 Java 소스코드가 있는 경로 설정
(기본경로는 'src/main/java')
- **xml.enabled**
추후 Jenkins plugin을 통해 coding convention 확인 결과를
받아 보기 위해 true로 설정
- **html.enabled**
html 형식의 보고서를 받아보지 않기 위해 false로 설정

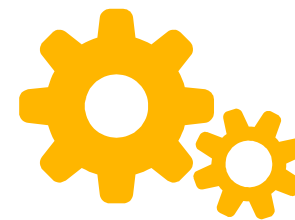


PMD

pmd.github.io

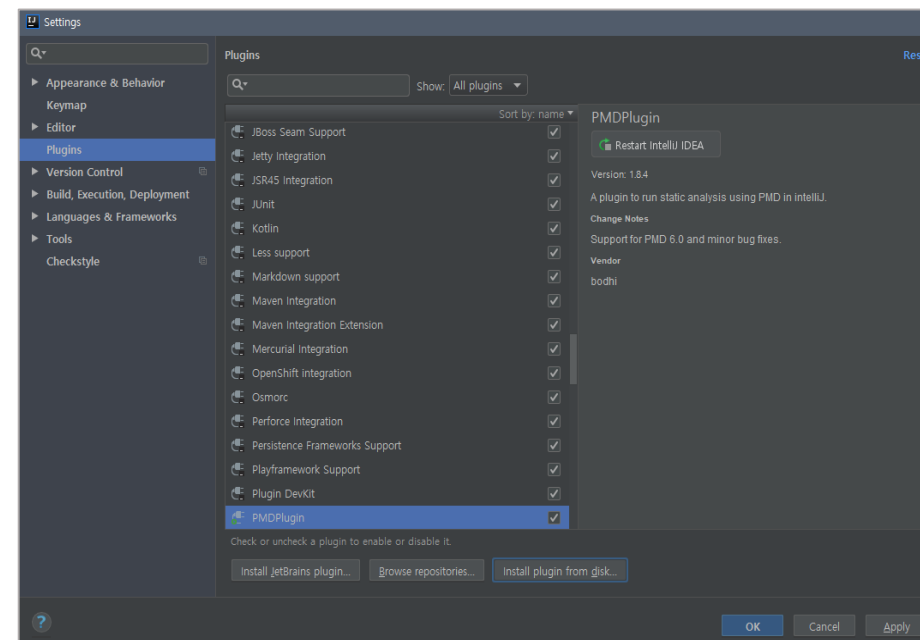
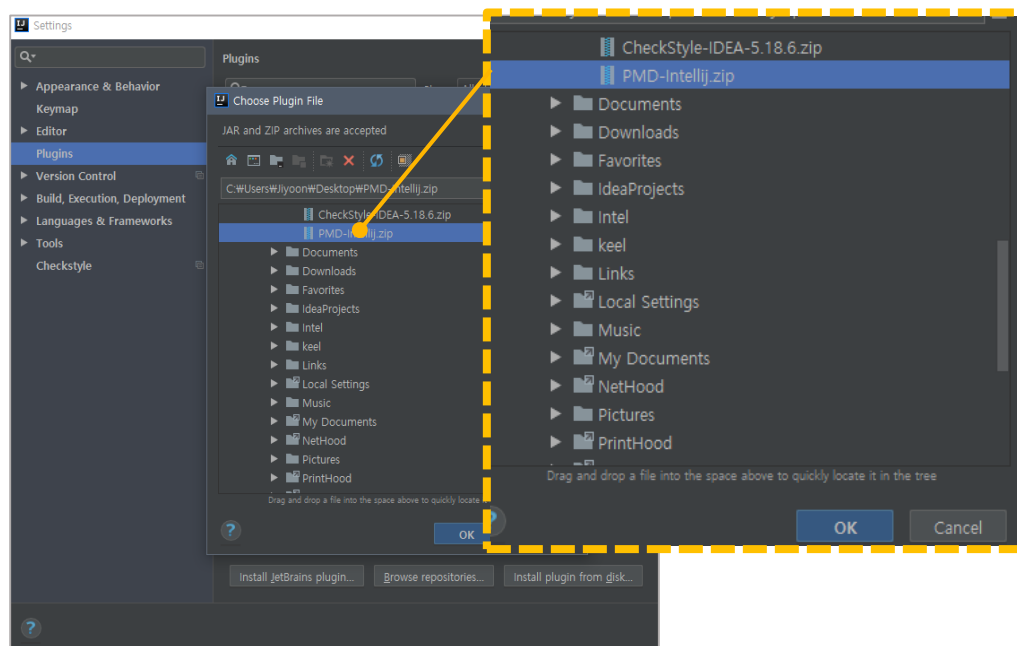
- 미사용 변수, 비어있는 코드 블록, 불필요한 오브젝트 생성과 같은 결함을 유발할 수 있는 코드를 검사해주는 도구
- Java 이외에도 다양한 프로그래밍 언어를 지원



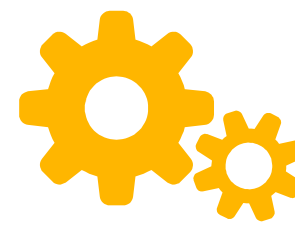


PMD

1) IntelliJ IDEA plugin 설치



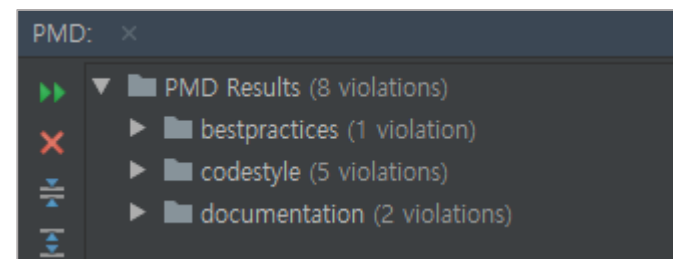
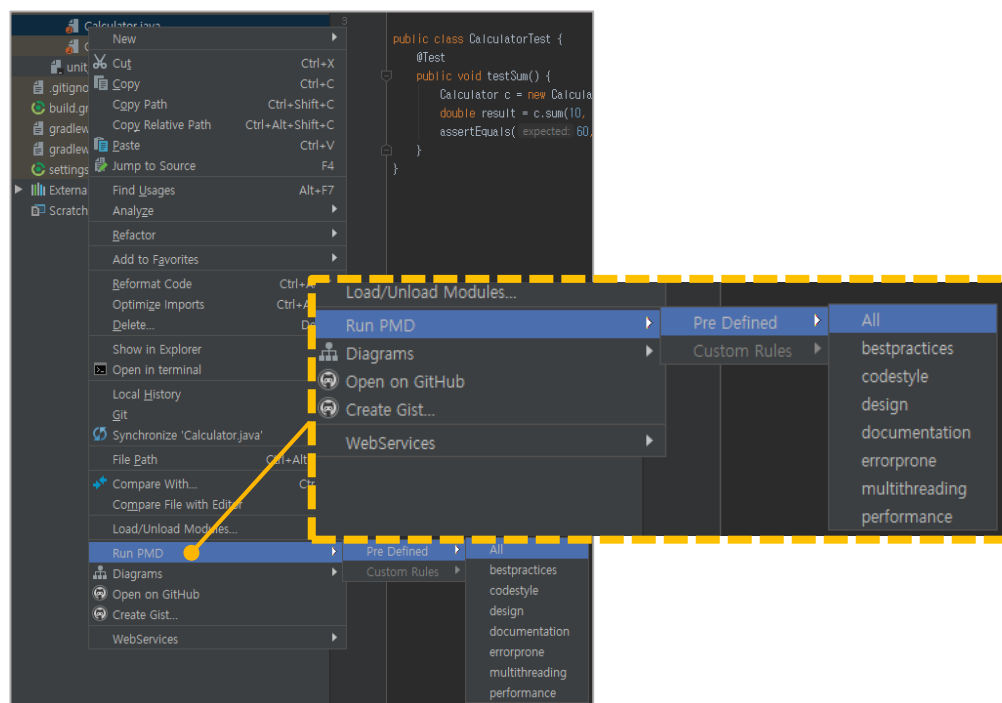
플러그인 설치: plugins.jetbrains.com/plugin/1137-pmdplugin

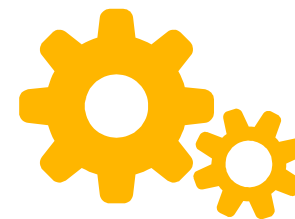


PMD

2) IntelliJ IDEA plugin 사용

Java 소스코드를 선택하고, PMD를 실행하여 결과 확인





PMD

3) Gradle script 작성

```
2  apply plugin: 'pmd'
3
4  pmd {
5      ignoreFailures = true
6      ruleSets = [
7          'java-basic',
8          'java-braces',
9          'java-clone',
10         'java-codesize',
11         'java-comments',
12         'java-controversial',
13         'java-coupling',
14         'java-design',
15         'java-empty',
16         'java-finalizers',
17         'java-imports',
18         'java-optimizations',
19         'java-strictexception',
20         'java-strings',
21         'java-typeresolution',
22         'java-unnecessary',
23         'java-unusedcode'
24     ]
25     sourceSets {
26         main {
27             java {
28                 srcDir 'unit_test/src'
29             }
30             resources {
31                 srcDir 'unit_test/src'
32             }
33         }
34     }
35 }
36
37 pmdMain {
38     reports {
39         xml.enabled = true
40         html.enabled = false
41     }
42 }
```

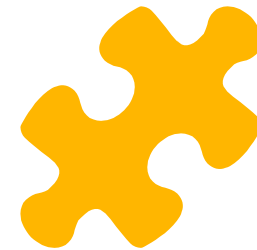
build.gradle

- **ruleSets**
PMD를 통해 검사할 규칙들에 대해서 정의
- **sourceSets**
분석할 Java 소스코드가 있는 경로 설정
(기본경로는 'src/main/java')
- **xml.enabled**
추후 Jenkins plugin을 통해 결과를 받아 보기 위해 true로 설정
- **html.enabled**
html 형식의 보고서를 받아보지 않기 위해 false로 설정

FindBugs

findbugs.sourceforge.net

- 현재 공식적으로 FindBug 프로젝트는 중단되고, SpotBugs로 전환됨
참고: github.com/findbugsproject/findbugs
- Byte code에서 버그 패턴을 찾아냄
- 100여개의 잠재적인 에러 타입을 찾아주며,
이는 scariest, scary, troubling, concern으로 구분되어 점수(rank)가 주어짐

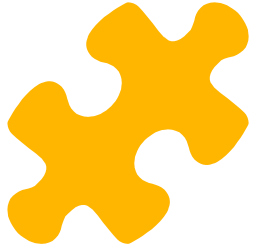


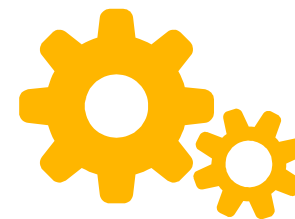
SpotBugs

spotbugs.github.io

- FindBugs의 후속 프로젝트
- 아직 공식적으로 Jenkins에서 SpotBugs Plugin을 지원하지는 않음

참고: issues.jenkins-ci.org/browse/JENKINS-46975





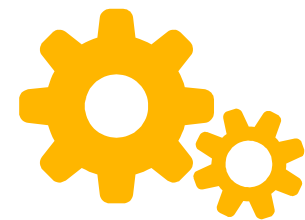
FindBugs

1) Gradle script 작성

```
1 group 'test'
2 version '1.0-SNAPSHOT'
3
4 apply plugin: 'java'
5 apply plugin: 'findbugs'
6
7 findbugs {
8     ignoreFailures = true
9 }
10
11 findbugsMain {
12     reports {
13         xml.enabled = true
14         html.enabled = false
15     }
16 }
17
18 sourceCompatibility = 1.8
19
20 repositories {
21     mavenCentral()
22 }
23
24 dependencies {
25     testCompile group: 'junit', name: 'junit', version: '4.12'
26 }
```

build.gradle

- **sourceSets**
분석할 Java 소스코드가 있는 경로 설정
(기본경로는 'src/main/java')
- **xml.enabled**
추후 Jenkins plugin을 결과를 받아 보기 위해 true로 설정
- **html.enabled**
html 형식의 보고서를 받아보지 않기 위해 false로 설정



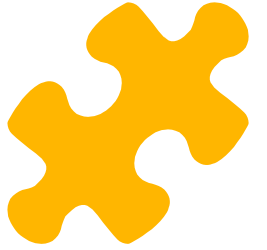
FindBugs

2) 빌드 결과 확인

```
C:\Users\Jiyeon\IdeaProjects\untitled2>gradlew clean build

> Task :findbugsMain
FindBugs rule violations were found. See the report at: file:///C:/Users/Jiyeon/IdeaProjects/untitled2/build/reports/findbugs/main.xml

BUILD SUCCESSFUL in 2s
4 actionable tasks: 4 executed
```

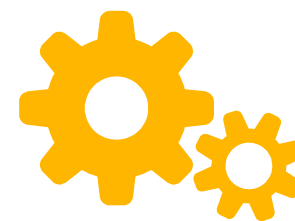


Error Prone

github.com/google/error-prone

- Compile-time에 흔한 프로그래밍 실수를 찾아주는 Java용 정적 분석 도구
- FindBugs와는 다르게, 컴파일러의 확장으로써 동작
- 구글의 Java 빌드 시스템에서 실제로 사용 중인 도구이며, 오픈소스로 공개함
 1. Hooks into your standard build, so all developers run it without thinking
 2. Tells you about mistakes immediately after they're made
 3. produces suggested fixes, allowing you to build tooling on it

관련 구글 논문: static.googleusercontent.com/media/research.google.com/en//pubs/archive/38275.pdf

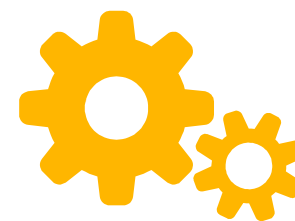


Error Prone

1) IntelliJ IDEA plugin 설치

The screenshot shows the IntelliJ IDEA interface with the 'Plugins' tab selected in the left sidebar. The 'Browse Repositories' dialog is open, displaying search results for 'Error-prone Compiler Integration'. The search bar contains 'Error-prone Compiler Integration' and the results list shows the plugin with 38,887 downloads and a 5-star rating. The 'Install' button is highlighted with a yellow dashed box. A yellow arrow points from the search bar to the selected plugin entry.

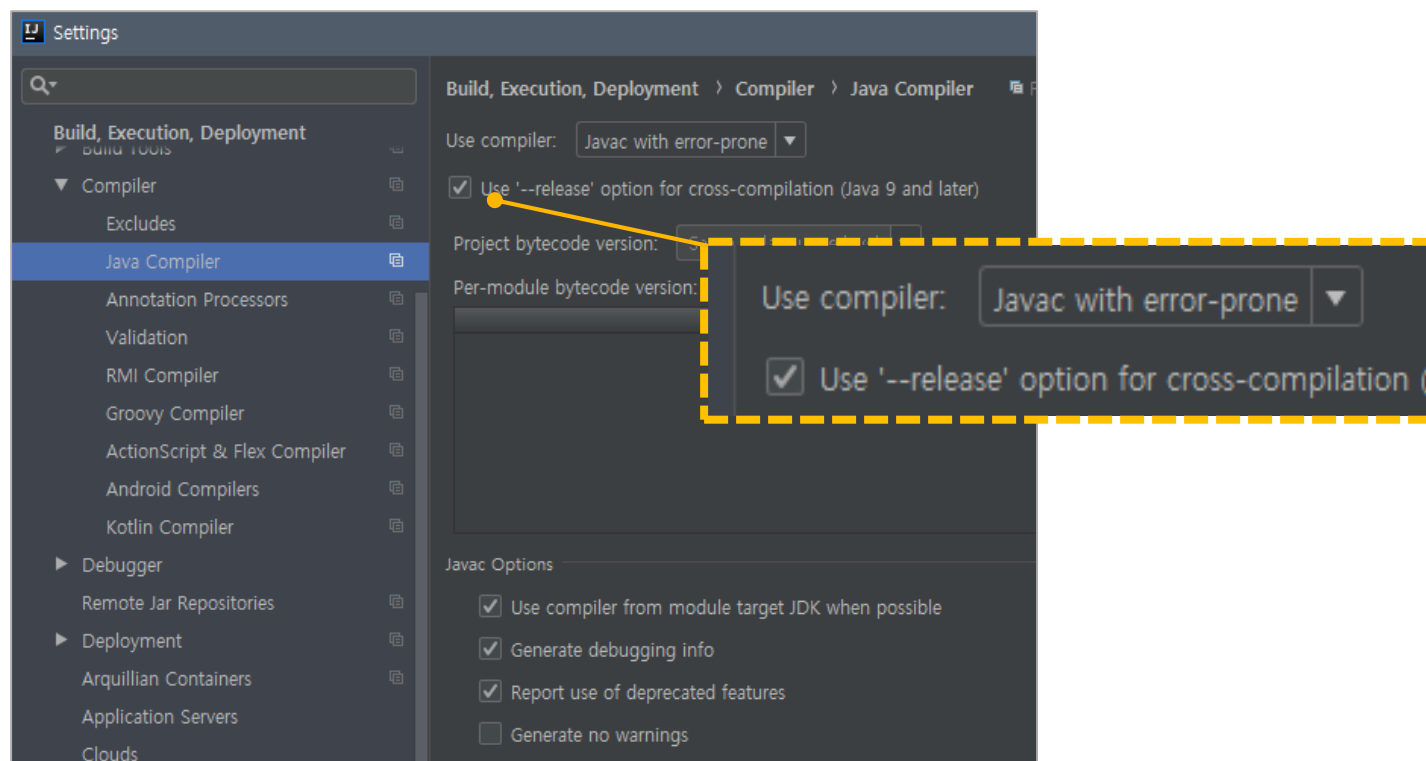
Plugin Name	Downloads	Rating	Updated	Version
Error-prone Compiler Integration	38,887	★★★★★	one month ago	v181.3986.9

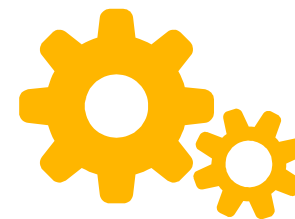


Error Prone

2) IntelliJ IDEA plugin 설정

Compiler 사용 설정을 'Javac'에서 'Javac with error-prone'으로 설정





Error Prone

3) IntelliJ IDEA plugin 사용

Build 실행 시, Error Prone이 프로그래밍 실수를 찾아주는 것을 확인

```
1 import java.util.HashSet;
2 import java.util.Set;
3
4 public class test
5 {
6     public static void main(String args[])
7     {
8         Set<Short> s = new HashSet<>();
9         for (short i = 0; i < 100; i++) {
10             s.add(i);
11             s.remove(i - 1);
12         }
13         System.out.println(s.size());
14     }
15 }
16
```

Messages: Build ×

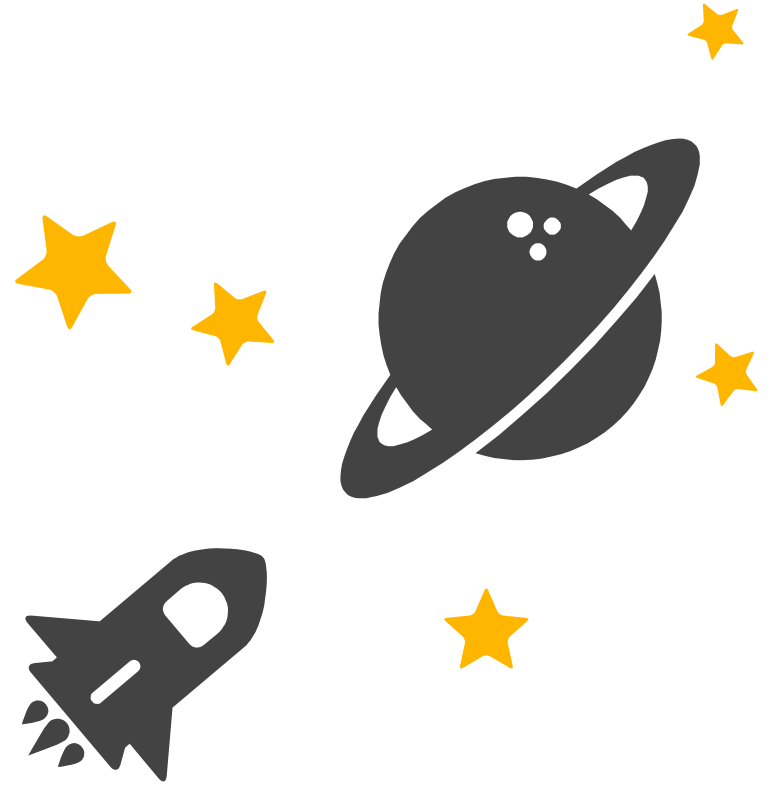
- Information: java: Errors occurred while compiling module 'untitled'
- Information: error-prone compiler 2.0.19 was used to compile java sources
- Information: 2018-04-13 오전 1:26 - Compilation completed with 1 error and 0 warnings in 2s 250ms

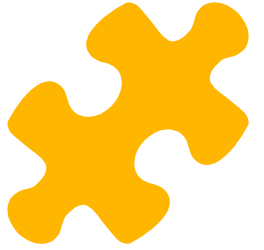
C:\Users\Wjyoon\IdeaProjects\untitled\src\test.java

Error:(11, 21) java: [CollectionIncompatibleType] Argument 'i - 1' should not be passed to this method; its type int is not compatible with its collection's type argument Short (see <http://errorprone.info/bugpattern/CollectionIncompatibleType>)

System testing

System testing은 각 모듈이 원래 계획했던 대로 작동하는지, 시스템의 실제 동작과 원래 의도했던 요구사항과는 차이가 없는지 등을 확인하는 과정





TestLink

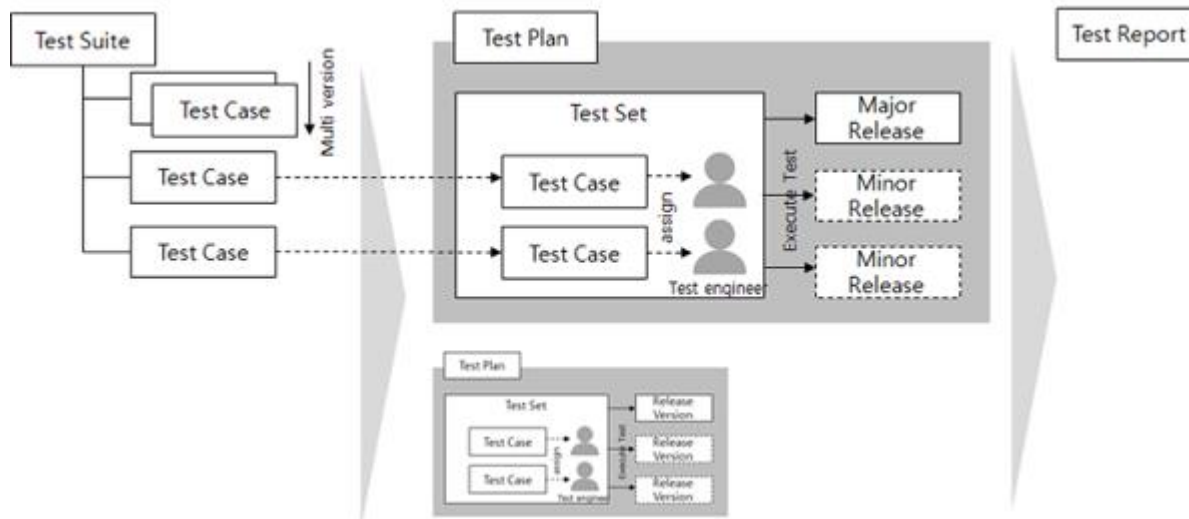
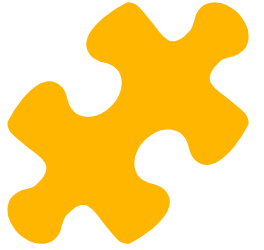
testlink.org

- 웹 기반으로 테스트 케이스 / 요구사항 관리도구
- 테스트 명세서와 계획, 리포팅, 요구사항 추적 기능 등을 제공
- 테스트 우선순위 부여, 사용자 활동 영역 정의, 트리 구조를 이용한 테스트 케이스 작성이 특징
- 다양한 종류의 DB 지원

Microsoft Azure cloud를 사용하여 환경 구축

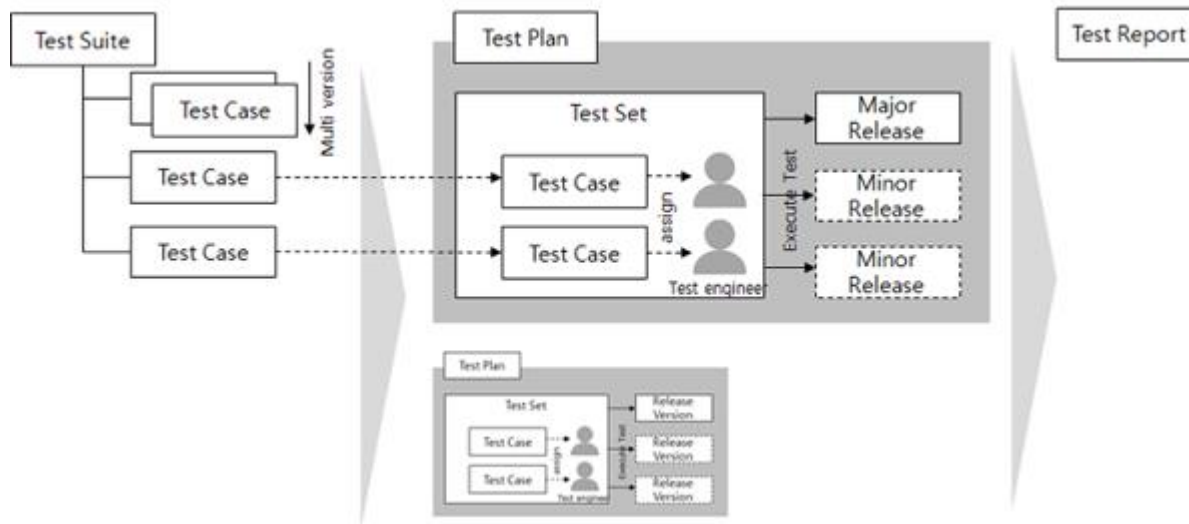
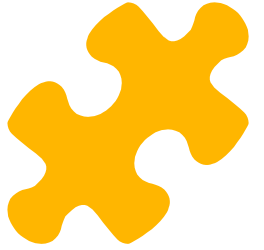


TestLink 개념 소개 - 1

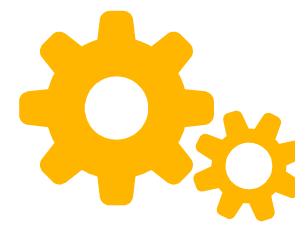


- **Test Project**
테스트를 수행하는 프로젝트 자체.
ex) 블로그 서비스 프로젝트, 카페 웹 서비스 테스트, ..
- **Test Specification**
테스트를 진행하고자 하는 test case들의 집합.
 - *Test suite: 대분류의 테스트 케이스*
 - *Test case: Test suite의 소분류의 테스트 케이스*
- **Test Plan**
실제 진행하는 테스트.
test spec 내의 test suite/case 들을 모아 놓은 것.
test engineer를 할당할 수 있음.

TestLink 개념 소개 - 2



- **Test Execution**
Test engineer가 할당된 test를 수행하고, Pass/Fail 여부를 확인.
- **Test Report**
Test 결과를 리포팅.
Test의 성공/실패, 주요 test suite의 성공/실패 등을 리포팅



TestLink

1) Project 생성

TestLink user [admin] TestLink 1.9.16 (Moka pot) Test Project

Test Project Management : Create a new project

Name * ctip-yeah

Prefix (used for Test case ID) * test

Project description

ctip-yeah testing

Enhanced features

- Enable Requirements feature
- Enable Testing Priority
- Enable Test Automation (API keys)
- Enable Inventory

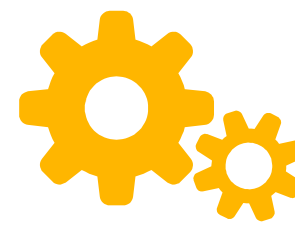
Issue Tracker Integration

>> There are no Issue Tracker Systems defined <<

Availability

- Active
- Public

Create Cancel



TestLink

2) 요구사항 작성

TestLink user [admin]

System

- Define Custom Fields
- Issue Tracker Management

Test Project

- Test Project Management
- Assign User Roles
- Assign Custom Fields
- Keyword Management
- Platform Management

Requirement Specification

- Requirement Specification**
- Requirement Overview
- Search Requirements
- Search Requirement Specifications
- Assign Requirements
- Requirement Monitoring Overview
- Generate Requirement Specification Document

Test Specification

- Test Specification
- Test Cases created per User

TestLink user [admin] TestLink 1.9.16 (Moka pot) Test Project test.ctip-year

Navigator - Requirement Specifications

Settings

- Update tree after every operation

Filters

- Document ID
- Title
- Status [Any]
- Requirement Type [Any]
- Req. Spec. Type [Any]
- Expected no. of testcases
- Has relation of type [Any]
- Test Case ID test-

Apply Reset Filters Advanced Filters

Expand tree Collapse tree

ctip-year (0)

Create Requirements Specification Test Project : ctip-year

Save Cancel

Document ID requirement-1

Title 요구사항-1: 덧셈

Scope

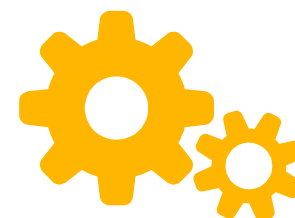
Normal B I U X₂ X₂ [Rich Text Editor Icons]

덧셈 수행 기능 필요

body p

Type **System Requirement Specification**

Save Cancel



TestLink

3) 요구사항에 관한 Operation 추가

Req. Spec. : [requirement-1] :: 요구사항-1: 덧셈

Requirement Specification Operations

Create Edit Delete Import Export Freeze all Requirements Create a new revision View history Print view

Requirement Operations

Create Import Export Create From Issues (XML) Create Test Cases Copy Requirements Bulk Monitoring

Req. Spec. : [requirement-1] :: 요구사항-1: 덧셈

revision : 2

Type : System Requirement Specification

Scope

덧셈 수행 기능 필요

Created on 04/12/2018 09:56:44 by user

Attached files :

Req. Spec. : 요구사항-1: 덧셈 - Create Requirement

Save Cancel

check to create another requirement after saving

Document ID
requirement-1_1

Title
덧셈 수행

Scope

Normal B I U X x²

덧셈 수행

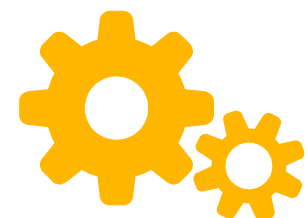
body p

Status Draft

Type System Function

Number of test cases needed 1

Save Cancel



TestLink

4) Test suite 작성

TestLink user [admin]

System
Define Custom Fields
Issue Tracker Management

Test Project
Test Project Management
Assign User Roles
Assign Custom Fields
Keyword Management
Platform Management

Requirement Specification
Requirement Specification
Requirement Overview
Search Requirements
Search Requirement Specifications
Assign Requirements
Requirement Monitoring Overview
Generate Requirement Specification Document

Test Specification
Test Specification
Test Cases Created per User

TestLink user [admin] test-

Navigator - Test Specification

Settings
Update tree after every operation

Filters
Test Case ID test-
Test Case Title
Status Select Some Options
Importance Select Some Options
Execution type [Any]

Test Project : ctip-year

Test Suite Operations

Test Project Name
ctip-year

Description
ctip-year

Attached
Test Suite Name
덧셈 수행

File

Test Project : ctip-year

Create Test Suite

Save Cancel

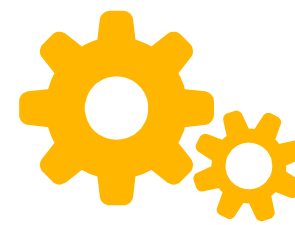
Test Suite Name
덧셈 수행

Details

Normal B I U X₂ X²

덧셈 수행


body p




TestLink

5) Test case 추가

Test Suite : 덧셈 수행

Test Suite Operations 

Test Case Operations 

Test Suite : 덧셈 수행

Details

덧셈 수행



Keywords : None

Test Suite : 덧셈 수행 - Create Test Case

Create Cancel check to create another test case after saving

Test Case Title
덧셈 수행



Summary


Normal - **B** *I* U ~~S~~ x₂ x² 

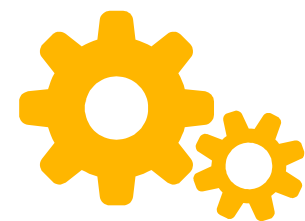
덧셈 수행

body p

Preconditions

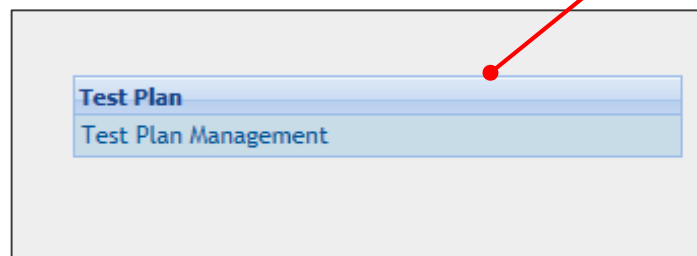

Normal - **B** *I* U ~~S~~ x₂ x² 

두 개의 수를 인자로 받는다



TestLink

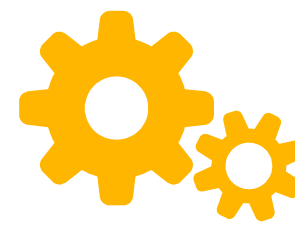
6) Test Plan 작성



Test Plan Management - Test Project ctip-year

Name	테스트 계획
Description	<p>Source</p> <p>Normal</p> <p>테스트 계획</p> <p>body p</p>
Active	<input checked="" type="checkbox"/>
Public	<input checked="" type="checkbox"/>

Create Cancel



TestLink

7) Test case에 요구사항 할당

TestLink user [admin]

System
Define Custom Fields
Issue Tracker Management

Test Project
Test Project Management
Assign User Roles
Assign Custom Fields
Keyword Management
Platform Management

Requirement Specification
Requirement Specification
Requirement Overview
Search Requirements
Search Requirement Specifications
Assign Requirements
Requirement Monitoring Overview
Generate Requirement Specification Document

Test Specification
Test Specification
Test Cases created per User

Test Case : 덧셈 수행

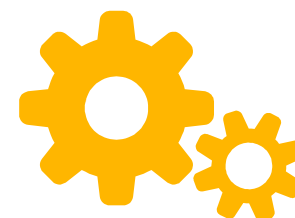
Assign Requirements to Test Case

Requirements Specification Document [requirement-1] - 요구사항-1: 덧셈

Assigned Requirements

<input checked="" type="checkbox"/>	Document ID	Requirement	Scope	Assigned by	Timestamp
<input type="checkbox"/>	requirements-1_1	덧셈 수행	덧셈 수행	user	04/12/2018 10:13:05

Unassign



TestLink

8) Test plan에 test case 추가

Test Plan
Test Plan Management
Builds / Releases
Milestone Overview

Test Execution
Execute Tests
Test Cases Assigned to Me
Test Reports and Metrics
Metrics Dashboard

Test Plan contents
~~Add / Remove Platforms~~
Add / Remove Test Cases
Assign Test Case Execution
Set Urgent Tests
Update Linked Test Case Versions
Show Test Cases Newest Versions

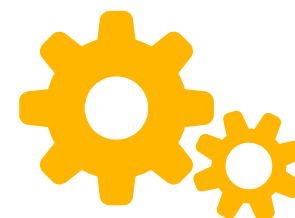
Test Plan : 테스트 계획 - Add/Remove Test Cases to/from Test Plan ?

Check/uncheck all Test cases for

Test cases with Latest Version with status with one of these values (Obsolete,Future) will not be displayed in this screen

덧셈 수행

<input checked="" type="checkbox"/> Test Case	Version	Importance			
<input checked="" type="checkbox"/> test-1 : 덧셈 수행	1	Medium	10000	<input type="checkbox"/>	04/12/2018
<input type="checkbox"/> test-2 : 두 개의 수를 인자로 받기	1	Medium	10010		



TestLink

9) Engineer에게 test case 작업 할당

Test Plan
Test Plan Management
Builds / Releases
Milestone Overview

Test Execution
Execute Tests
Test Cases Assigned to Me
Test Reports and Metrics
Metrics Dashboard

Test Plan contents
Add / Remove Platforms
Add / Remove Test Cases
Assign Test Case Execution
Set Urgent Tests
Update Linked Test Case Versions
Show Test Cases Newest Versions

덧셈 수행

<input checked="" type="checkbox"/>	Test Case [Version]	Priority	Assigned to	Assign to
<input checked="" type="checkbox"/>	test-1 : 덧셈 수행 [1]	Medium		<input type="text" value="user x"/>

Test Project: ctip-year - Test Cases Assigned to User: user

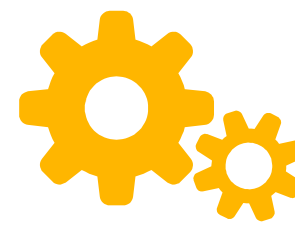
Show also closed builds

Test Plan: 테스트 계획

Expand/Collapse Groups Show all Columns Reset to Default State Refresh Reset Filters | MultiSort

Test Suite	Test Case
Build: first test (1 Item)	
덧셈 수행	test-1 : 덧셈 수행 [v1]

Generated by TestLink on 04/12/2018 10:17:21



TestLink

10) Build 생성

Test Plan

- Test Plan Management
- Builds / Releases**
- Milestone Overview

Test Execution

- Execute Tests
- Test Cases Assigned to Me
- Test Reports and Metrics
- Metrics Dashboard

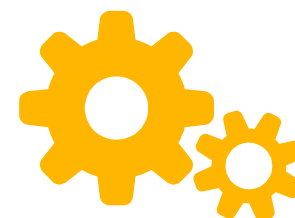
Test Plan contents

- Add / Remove Platforms
- Add / Remove Test Cases
- Assign Test Case Execution
- Set Urgent Tests
- Update Linked Test Case Versions
- Show Test Cases Newest Versions

Build management - Test Plan : 테스트 계획

Title	Description
build-1	first build

Create



TestLink 11) Test 실행

Test Plan
Test Plan Management
Builds / Releases
Milestone Overview

Test Execution
Execute Tests
Test Cases Assigned to Me
Test Reports and Metrics
Metrics Dashboard

Test Plan contents
Add / Remove Platforms
Add / Remove Test Cases
Assign Test Case Execution
Set Urgent Tests
Update Linked Test Case Versions
Show Test Cases Newest Versions

Last execution (any build)
Date : 04/12/2018 10:23:12 - Tested by : user - Build : build-1 - Status : Passed

Last execution (current build) - Build : build-1

Date	Build	Tested by	Status	Exec (min)	Version	Run mode
04/12/2018 10:23:12	build-1	user	Passed		1	

Test Case test-1 :: Version : 1 :: 맞셈 수행
Assigned to : user

Summary
맞셈 수행

Preconditions
두 개의 수를 인자로 받는다

#	Step actions	Expected Results	Execution	Execution notes	Result
1	a + b	a + b	Automated	test done	Passed

File

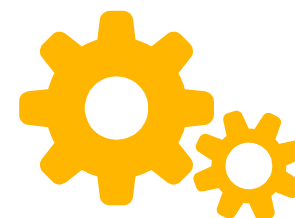
Execution type : Automated
Estimated exec. duration (min) :

Requirements
[요구사항-1: 맞셈] requirement-1_1 : 맞셈 수행

Notes / Description

Execution duration (min)

Move to Next Test Case



TestLink

12) Test report

Test Plan
Test Plan Management
Builds / Releases
Milestone Overview

Test Execution
Execute Tests
Test Cases Assigned to Me
Test Reports and Metrics
Metrics Dashboard

Test Plan contents
Add / Remove Platforms
Add / Remove Test Cases
Assign Test Case Execution
Set Urgent Tests
Update Linked Test Case Versions
Show Test Cases Newest Versions

TestLink user [admin]

Printed by TestLink on 04/12/2018
2012 © TestLink Community

Test Project: ctip-yearh

ctip-yearh testing

Test Plan: 테스트 계획

테스트 계획

Test Suite : 덧셈 수행

Reports and Metrics

Report Format HTML Print

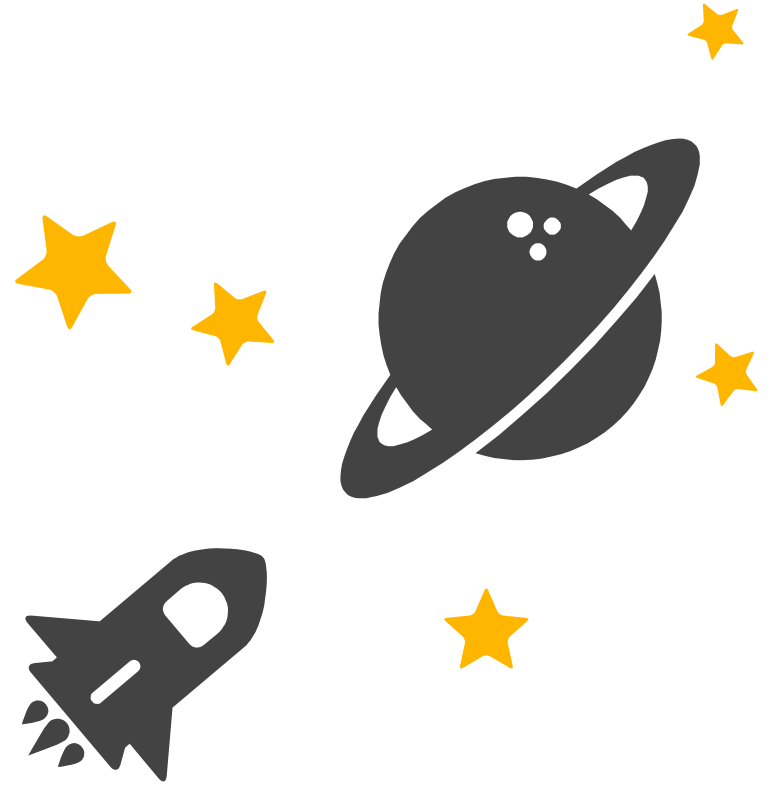
Test Plan 테스트 계획

Show inactive test plans

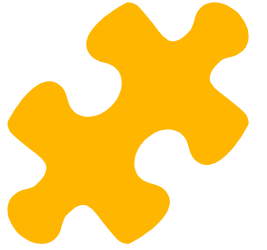
- Test Plan Report
- Test Report
- Test Report on build
- General Test Plan Metrics
- Results by Tester per Build
- Test Case Assignment Overview
- Test result matrix
- Test Results Flat
- Failed Test Cases
- Blocked Test Cases
- Not run Test Cases
- Test Cases without Tester Assignment
- Charts
- Requirements based Report
- Test Cases with Custom Fields set on Execution
- Test Plan with Custom Field info
- Test Cases not assigned to Any Test Plan

CTIP Environment

정적 분석 도구들과 Jenkins의 연동 및 CTIP 환경 정리



CTIP Environment



- Sonarqube - Jenkins 연동
- Jenkins - PMD, CheckStyle, FindBugs Report 연동



Sonarqube - Jenkins 연동

1) Jenkins > Plugin Manager에서 'SonarQube Scanner' 설치

The screenshot shows the Jenkins Plugin Manager interface. The search filter is set to 'sonar'. The '설치 가능' (Installable) tab is active, displaying a list of plugins. The 'SonarQube Scanner' plugin is checked for installation. A yellow dashed box highlights the 'SonarQube Scanner' plugin in the list, and a yellow arrow points to its checkbox. The interface also shows options to '재시작 없이 설치하기' (Install without restart) or '지금 다운로드하고 재시작 후 설치하기' (Download now and install after restart), along with a '지금 확인' (Check now) button. The update information indicates it was obtained 21 seconds ago.

설치	이름	버전
<input type="checkbox"/>	CodeSonar	2.0.6
<input checked="" type="checkbox"/>	SonarQube Scanner	2.7.1
<input type="checkbox"/>	Sonargraph Integration	2.1.2
<input type="checkbox"/>	Sonargraph	1.6.4
Mashup Portlets		
<input type="checkbox"/>	Additional Dashboard Portlets (shows the SCM changes for Test Results Portlet (shows the	
<input type="checkbox"/>	Sonar Gerrit	
Quality Gates		
<input type="checkbox"/>	Fails the build whenever the Quality Gates criteria in the Sonar 5.6+ analysis aren't met (the project Quality Gates status is different than "Passed")	
Sonar Quality Gates		
<input type="checkbox"/>	Fails the build whenever the Quality Gates criteria in the Sonar 5.6+ analysis aren't met (the project Quality Gates status is different than "Passed")	1.3.0



Sonarqube - Jenkins 연동

2) Jenkins > Global Tool Configuration에서 'SonarQube Scanner' 설정

The screenshot shows the Jenkins Global Tool Configuration page for SonarQube Scanner. The page is titled "Jenkins > Global Tool Configuration" and has a sub-header "SonarQube Scanner for MSBuild". Below this, there is a section for "SonarQube Scanner for MSBuild installations" with an "Add SonarQube Scanner for MSBuild" button. A list of installations is shown, with one entry for "SonarQube Scanner". The entry has a name field set to "Sonar Runner", a "Required" status, and a checked "Install automatically" checkbox. Below this, there is a section for "Install from Maven Central" with a version dropdown set to "SonarQube Scanner 3.1.0.1141". At the bottom of the list, there are buttons for "Delete Installer" and "Delete SonarQube Scanner".



Sonarqube - Jenkins 연동

3) Sonarqube > My Account > Security에서 Token 발급

The screenshot shows the SonarQube web interface. The top navigation bar includes the SonarQube logo and links for Projects, Issues, Rules, Quality Profiles, Quality Gates, and Administration. A search bar is present on the right. The user is logged in as 'Administrator'. The main content area is titled 'Tokens' and contains the following text:

Tokens

If you want to enforce security by not providing credentials of a real SonarQube user to run your code scan or to invoke web services, you can provide a User Token as a replacement of the user login. This will increase the security of your installation by not letting your analysis user's password going through your network.

Generate Tokens

Enter Token Name

Name	Created	
jenkins	April 13, 2018	<input type="button" value="Revoke"/>



Sonarqube - Jenkins 연동

4) Jenkins > Global Tool Configuration에서 'SonarQube servers' 설정

SonarQube servers

Environment variables Enable injection of SonarQube server configuration as build environment variables
If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

SonarQube installations

Name

Server URL
Default is http://localhost:9000

Server authentication token
SonarQube authentication token. Mandatory when anonymous access is disabled.

Sonarqube에서 발급받은 Token 입력

List of SonarQube installations



Sonarqube - Jenkins 연동

5) Jenkins Project Configure에서 Build 설정

Build

Invoke Gradle script

Invoke Gradle

Gradle Version: Gradle 4.6

Use Gradle Wrapper

Tasks: clean build

Execute SonarQube Scanner

Task to run: []

JDK: (Inherit From Job)

Path to project properties: []

Analysis properties: sonar.projectKey=ctip-yeah
sonar.projectName=ctip-yeah
sonar.sources=unit_test/src/
sonar.sourceEncoding=UTF-8
sonar.language=java
sonar.java.binaries=build

Additional arguments: []

JVM Options: []

저장 Apply

Tasks

clean build

Analysis properties

```
sonar.projectKey=ctip-yeah
sonar.projectName=ctip-yeah
sonar.sources=unit_test/src/
sonar.sourceEncoding=UTF-8
sonar.language=java
sonar.java.binaries=build
```



Sonarqube - Jenkins 연동

6) 연동 결과 확인

Back to Dashboard

Status

Changes

Workspace

Build Now

Delete Project

Configure

GitHub Hook Log

GitHub

SonarQube

Project ctip-yearh

SonarQube

Workspace

Recent Changes

SonarQube Quality Gate

ctip-yearh **OK**

server-side processing: **Success**

Permalinks

- Last build (#37), 18 min ago
- Last stable build (#37), 18 min ago

Build History

Build ID	Time
#37	Apr 14, 2018 1:03 PM
#36	Apr 14, 2018 11:49 AM
#35	Apr 14, 2018 11:40 AM
#34	Apr 14, 2018 8:25 AM



Jenkins - 각종 Plugin 연동

1) Jenkins > Plugin Manager에서 아래의 plugin들을 설치

- [FindBugs Plug-in](#)
This plug-in collects the [FindBugs](#) analysis results of the project modules and visualizes the found warnings.
- [Checkstyle Plug-in](#)
This plug-in collects the [Checkstyle](#) analysis results of the project modules and visualizes the found warnings.
- [PMD Plug-in](#)
This plug-in collects the [PMD](#) analysis results of the project modules and visualizes the found warnings.



Jenkins - 각종 Plugin 연동

2) build.gradle 작성 (예시)

1

```
group 'test'
version '1.0-SNAPSHOT'

apply plugin: 'java'
apply plugin: 'checkstyle'
apply plugin: 'pmd'
apply plugin: 'findbugs'

checkstyle {
    ignoreFailures = true
    configFile = file('config/checkstyle/checkstyle.xml')
    toolVersion = '8.8'
    sourceSets {
        main {
            java {
                srcDir 'unit_test/src'
            }
        }
    }
}
```

2

```
pmd {
    ignoreFailures = true
    ruleSets = [
        'java-basic', 'java-braces',
        'java-clone', 'java-codesize',
        'java-comments', 'java-controversial',
        'java-coupling', 'java-design',
        'java-empty', 'java-finalizers',
        'java-imports', 'java-optimizations',
        'java-strictexception', 'java-strings',
        'java-typeresolution', 'java-unnecessary',
        'java-unusedcode'
    ]
    sourceSets {
        main {
            java {
                srcDir 'unit_test/src'
            }
        }
    }
}
```

3

```
checkstyleMain {
    reports {
        xml.enabled = true
        html.enabled = false
    }
}

pmdMain {
    reports {
        xml.enabled = true
        html.enabled = false
    }
}

findbugsMain {
    reports {
        xml.enabled = true
        html.enabled = false
    }
}
```

4

```
sourceCompatibility = 1.8

repositories {
    mavenCentral()
}

dependencies {
    compile 'junit:junit:4.12'
}
```



Jenkins - 각종 Plugin 연동

3) Jenkins Project Configure에서 Build 설정

Post-build Actions

Publish Checkstyle analysis results

Checkstyle results

[Fileset includes](#) setting that specifies the generated raw CheckStyle XML report files, such as `**/checkstyle-result.xml`. Basedir of the fileset is [the workspace root](#). If no value is set, then the default `**/checkstyle-result.xml` is used. Be sure not to include any non-report files into this pattern.

Publish FindBugs analysis results

FindBugs results

[Fileset includes](#) setting that specifies the generated raw FindBugs XML report files, such as `**/findbugs.xml` or `**/findbugsXml.xml`. Basedir of the fileset is [the workspace root](#). If no value is set, then the default `**/findbugsXml.xml` or `**/findbugs.xml` are used for maven or ant builds, respectively. Be sure not to include any non-report files into this pattern.

Use rank as priority

Uses the bug rank when evaluating the priority of the warnings (otherwise the FindBugs priority is used).

Publish PMD analysis results

PMD results

[Fileset includes](#) setting that specifies the generated raw PMD XML report files, such as `**/pmd.xml`. Basedir of the fileset is [the workspace root](#). If no value is set, then the default `**/pmd.xml` is used. Be sure not to include any non-report files into this pattern.

Checkstyle results

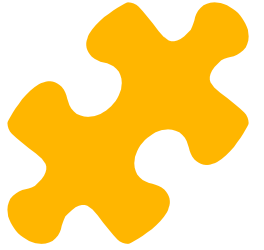


Jenkins - 각종 Plugin 연동

4) 연동 결과 확인

The screenshot displays the Jenkins interface for Build #37, dated April 14, 2018, at 1:03:11 PM. On the left sidebar, the 'Checkstyle Warnings' and 'PMD Warnings' links are highlighted with a red dashed box. The main content area shows the following details:

- Build #37 (Apr 14, 2018 1:03:11 PM)**
- Changes:** 1. " to ' ([commit: 1045f7c](#)) ([detail](#) / [githubweb](#))
- Started by:** [GitHub push by punkyoon](#) (3 times)
- Revision:** 1045f7c2317563784d528fe4af4c7e461f1c1ab1
 - [refs/remotes/origin/master](#)
- Checkstyle:** [10 warnings](#) from one analysis.
- FindBugs:** 0 warnings from one analysis.
 - No warnings since build 33.
 - New zero warnings highscore: no warnings since yesterday!
- PMD:** [10 warnings](#) from one analysis.



CTIP Environment

IntelliJ IDEA 2018.1 | Java 8 | JUnit 4 | Google convention

Build



CI



Jenkins

Issue 관리 / Code 관리



GitHub

System Test



TestLink

정적 분석



FindBugs



CheckStyle



PMD



Q & A



Thank you 😊