

Mantis, SVN & JUnit

Team Presentation #4

Team 2

201111344 김재엽

201111339 김민우

201211386 최하나

Contents

Chapter 1 – Mantis

Chapter 2 – Subversion

Chapter 3 – JUnit

Chapter 1

Mantis

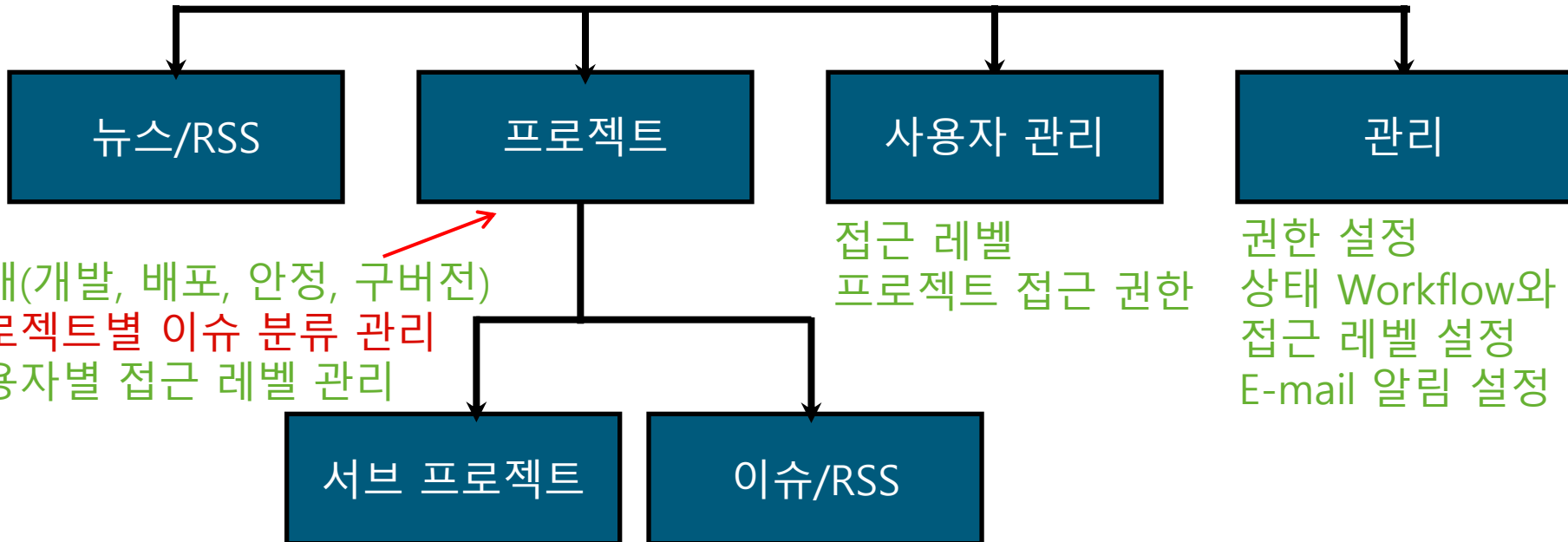
Mantis

What is Mantis?

- 프로젝트 단위로 이슈 관리
- 이슈는 분류, 중요도, 우선순위, 첨부 파일 등의 정보를 가짐
- 사용자에게 공지할 뉴스를 등록하여 공지
- 사용자 관리 (설정, 권한 등)
- E-mail 알림 기능
- Wiki와 연동
- 이슈 번호로 소스버전관리시스템이랑 연동

Mantis

Overview



Mantis

Access Level

보기 가능

- 보고자에게 이슈를 재 오픈할 수 있도록 한다.
- 핸들러 할당에 대한 상태를 설정한다.
- 사용자가 자기 이슈를 수정할 수 있도록 한다.
- 보기 변경 기록
- 보기 이슈 할당
- 보기 이슈 역사

보고 가능

- 이슈 보고
- 이슈 감시
- 보기 상태 설정(공개 또는 비공개)
- 이슈노트 추가
- 그 외의 보기 이슈의 기능을 포함한다.

Mantis

Access Level(cont.)

갱신 가능

- 이슈 갱신
- 보기 상태 바꾸기(공개 또는 비공개)
- 그 외엔 보고 가능한 기능을 포함한다.

개발자

- 이슈 다루기
- 이슈 할당
- 이슈 이동
- 이슈 삭제
- 이슈 재발생
- 이슈 상태 갱신
- 비공개 이슈 보기

매니저

- 읽기 전용 이슈 갱신
- 그 외의 개발자의 기능을 포함한다.

관리자

- 모든 기능을 관리할 수 있는 최고 관리자

Mantis

Access Level(cont.)

Viewer

- 버그와 버그리스트를 볼 수만 있다 (e.g. guest)

Reporter

- 새로운 버그를 보고할 수 있다

Updater

- 리포터와 권한이 같지만, 버그를 업데이트 할 수 있다

Developer

- updater보다 더 많은 권한을 가지고 있고, 실제 프로젝트에 대한 개발이 가능하다

Manager

- developers보다 더 많은 권한을 가지고 있고, 관리하고 있는 프로젝트들에 대해서 모든 권한을 가지고 있다.

Administrator

- 모든 권한을 가지고 있다. 계정 생성, 비밀번호 초기화, 유저 추가 삭제 등

Mantis

Mantis with Testing Team

- <http://dev.viskit.co.kr:20080/mantis>

Signup	
Username:	<input type="text"/>
E-mail:	<input type="text"/>
Enter the code as it is shown in the box on the right.:	<input type="text"/> 
<p>On completion of this form and verification of your answers, you will be sent a confirmation e-mail to the e-mail address you specified. Using the confirmation e-mail, you will be able to activate your account. If you fail to activate your account within seven days, it will be purged. You must specify a valid e-mail address in order to receive the account confirmation e-mail.</p>	
<input type="button" value="Signup"/>	

[[Login](#)] [[Lost your password?](#)]

Mantis

Create Project

- Mantis administrator 로그인
- 관리 -> 프로젝트 관리 -> 새로운 프로젝트 생성

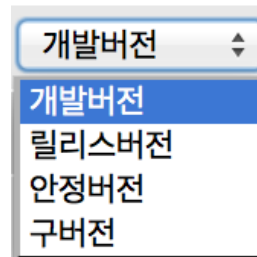
프로젝트 추가	
*프로젝트 이름	<input type="text" value="Test Project"/>
상태	<input type="text" value="개발버전"/>
전역 분류 상속	<input checked="" type="checkbox"/>
상태 보기	<input type="text" value="공개"/>
설명	<input type="text" value="테스트로 생성한 프로젝트"/>
<input type="button" value="프로젝트 추가"/>	

Mantis

Project

프로젝트 상태

- 개발버전
- 릴리즈버전
- 안정버전
- 구버전



서브 프로젝트

- 메인 프로젝트에서 여러 개의 하위 프로젝트를 생성할 수 있다.
- 하위 프로젝트에 관리자는 물론이며 익명의 사용자가 보는 것 까지 설정할 수 있다.

이름	상태	사용가능	분류 상속	상태 보기	설명	실행
Test Sub Project	개발버전	X	<input type="checkbox"/>	공개	Test Project에서 하위 프로젝트로 작업하는 프로젝트	[편집] [링크해제]
Test Sub Project 2	개발버전	X	<input type="checkbox"/>	공개	Test Project의 두 번째 하위 프로젝트	[편집] [링크해제]

하위 프로젝트 상속 관계 바꾸기

서브프로젝트로 추가

Mantis

Project(cont.)

프로젝트 분류

- 이슈를 등록할 때 해당 이슈에 대한 분류를 지정할 수 있다.

분류	분류	할당하기	실행
[모든 프로젝트] General			
버그			[편집] [삭제]
순탄한			[편집] [삭제]
지식			[편집] [삭제]
치명적			[편집] [삭제]
<input type="text"/>	<input type="button" value="분류 추가"/>		
<input type="button" value="카테고리 복사(From)"/> <input type="button" value="카테고리 복사(To)"/>			

프로젝트 버전

- 이슈를 해당 버전에 넣어서 관리한다.
- 어떤 이슈가 어느 버전에서 생기고, 어느 버전에서 고쳐지고 어느 버전을 목표 등으로 사용될 수 있다.

이름	상태	사용가능	분류 상속	상태 보기	설명	실행
Test Sub Project	개발버전	X	<input type="checkbox"/>	공개	Test Project에서 하위 프로젝트로 작업하는 프로젝트	[편집] [링크해제]
Test Sub Project 2	개발버전	X	<input type="checkbox"/>	공개	Test Project의 두 번째 하위 프로젝트	[편집] [링크해제]
<input type="button" value="하위 프로젝트 상속 관계 바꾸기"/>						
<input type="button" value="서브프로젝트로 추가"/>						

Mantis

My Page

- 이슈에 관련된 정보를 간략하게 보여준다.



로그인한 ID :: papimomi (메니저)

2013-04-18 16:08 KST

프로젝트: 모든 프로젝트 이동

[첫화면](#) | [내 페이지](#) | [이슈 보기](#) | [이슈 보고하기](#) | [변경 기록](#) | [로드맵](#) | [요약](#) | [관리](#) | [계정 관리](#) | [로그아웃](#)

이슈 번호 이동

할당되지 않음 [^] (1 - 2 / 2)

0000003	[MyBuildingServer] 클라이언트에서 방 생성시 오류
지영혁 - 2013-04-17 20:18	
0000002	[MyBuildingServer] 빌딩을 모두 파괴
기능추가 - 2013-04-17 20:09	

해결됨 [^] (0 - 0 / 0)

내가 모니터링 중 [^] (0 - 0 / 0)

내가 보고함 [^] (0 - 0 / 0)

최근에 변경됨 [^] (1 - 3 / 3)

0000003	[MyBuildingServer] 클라이언트에서 방 생성시 오류
지영혁 - 2013-04-17 20:18	
0000001	[MyBuildingServer] 새로운 식사를 추가할 필요
[모든 프로젝트] General - 2013-04-17 20:09	
0000002	[MyBuildingServer] 빌딩을 모두 파괴
기능추가 - 2013-04-17 20:09	

신규 정보 부족 이슈 검토 확인된 이슈 할당된 이슈 해결된 이슈 닫음

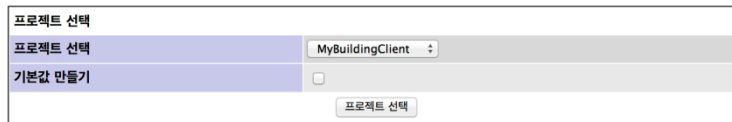
Copyright © 2000 - 2013 MantisBT Team
papimomi@braingyuys.net



Mantis

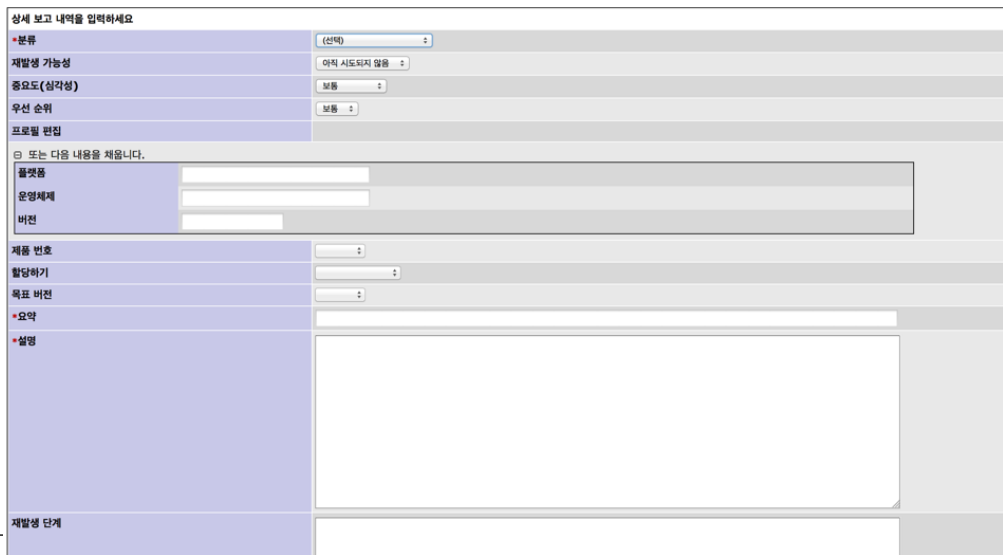
Issue

- Mantis에 등록된 프로젝트가 여러 개일 경우
- 이슈를 등록할 프로젝트를 선택



A screenshot of the Mantis project selection form. It features a '프로젝트 선택' (Project Selection) section with a dropdown menu currently showing 'MyBuildingClient'. Below the dropdown is a '기본값 만들기' (Create Default) checkbox and a '프로젝트 선택' (Project Selection) button.

- 이슈 보고를 하기 위해 필요한 빈칸을 채워주면 된다.



A screenshot of the Mantis issue reporting form. The form is titled '상세 보고 내역을 입력하세요' (Enter detailed report details). It includes several sections: '분류' (Category) with a dropdown menu, '개발성 가능성' (Development Possibility) with a dropdown menu, '중요도 (심각성)' (Priority (Severity)) with a dropdown menu, '우선 순위' (Priority) with a dropdown menu, and '프로젝트 편집' (Project Edit). Below these are several input fields for '플랫폼' (Platform), '운영체제' (OS), and '버전' (Version). There are also dropdown menus for '제품 번호' (Product Number), '할당하기' (Assign), and '목표 버전' (Target Version). The form also has sections for '요약' (Summary) and '설명' (Description) with text input areas. At the bottom, there is a '개발성 단계' (Development Stage) dropdown menu.

Mantis

Issue(cont.)

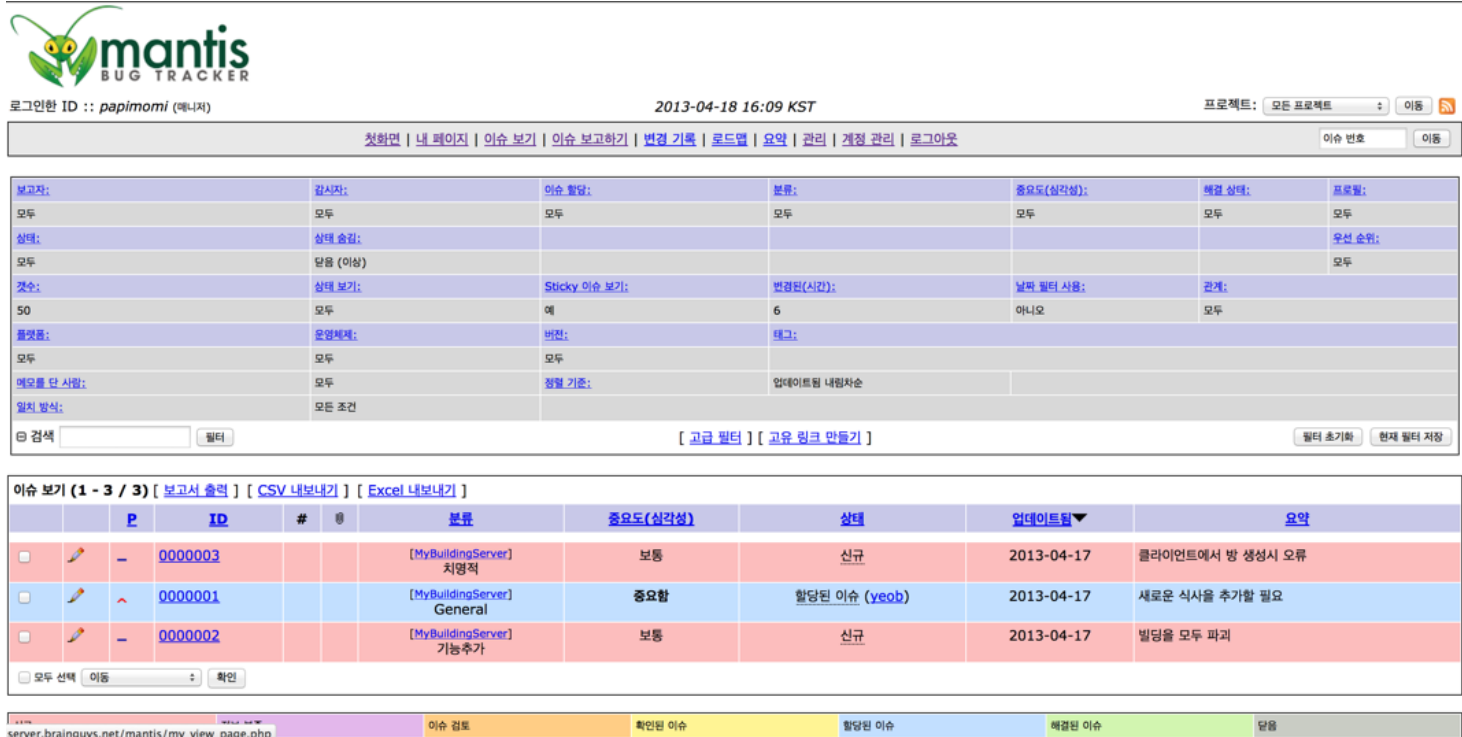
- 버전 관리를 한다면 다음과 같이 입력한다.
- 로드맵에서 진행상황을 볼 수 있다.
- 해당 이슈를 특정 개발자에게 할당할 수 있음

제품 번호	1.0.x
할당하기	AuraWalker
목표 버전	1.1.x
*요약	건물 스프라이트 변경
*설명	건물에 따라서 이미지를 만들어가지고 랜덤으로 보여질 수 있도록 기능을 추가

Mantis

Issue(cont.)

- 현재 등록된 이슈들을 볼 수 있다.



The screenshot shows the Mantis Bug Tracker interface. At the top left is the Mantis logo with the text 'mantis BUG TRACKER'. Below it, the user ID is '로그인한 ID :: papimomi (메니저)' and the current time is '2013-04-18 16:09 KST'. There are navigation links for '첫화면', '내 페이지', '이슈 보기', '이슈 보고하기', '변경 기록', '로드맵', '요약', '관리', '계정 관리', and '로그아웃'. A search bar and '이슈 번호' field are also present.

Below the navigation is a filter grid with various options:

보고자:	감시자:	이슈 할당:	분류:	중요도(심각성):	해결 상태:	프록시:
모두	모두	모두	모두	모두	모두	모두
상태:	상태 숨김:					우선 순위:
모두	닫음 (이상)					모두
갯수:	상태 보기:	Sticky 이슈 보기:	변경된(시간):	날짜 필터 사용:	관계:	
50	모두	예	6	아니오	모두	
플랫폼:	운영체제:	버전:	태그:			
모두	모두	모두				
메모를 단 사람:	모두	정렬 기준:	업데이트된 내림차순			
일치 방식:	모든 조건					

At the bottom of the filter grid are buttons for '검색', '필터', '[고급 필터]', '[고유 링크 만들기]', '필터 초기화', and '현재 필터 저장'.

Below the filter grid is the '이슈 보기 (1 - 3 / 3)' section with links for '[보고서 출력]', '[CSV 내보내기]', and '[Excel 내보내기]'. It contains a table of issues:

	P	ID	#	명	분류	중요도(심각성)	상태	업데이트됨	요약
<input type="checkbox"/>		-	0000003		[MyBuildingServer] 치명적	보통	신규	2013-04-17	클라이언트에서 방 생성시 오류
<input type="checkbox"/>		^	0000001		[MyBuildingServer] General	중요함	할당된 이슈 (yeob)	2013-04-17	새로운 식사를 추가할 필요
<input type="checkbox"/>		-	0000002		[MyBuildingServer] 기능추가	보통	신규	2013-04-17	빌딩을 모두 파괴

At the bottom of the issue list is a checkbox for '모두 선택' and a '확인' button.

At the very bottom of the screenshot is a navigation bar with buttons for '이슈 검토', '확인된 이슈', '할당된 이슈', '해결된 이슈', and '닫음'.

Mantis

Issue(cont.)

- 자신이 찾는 이슈를 필터링하여 화면에 출력할 수 있다.

보고자:	감시자:	이슈 할당:	분류:	중요도(심각성):	해결 상태:	프로필:
administrator ▾	[모두] ▾	모두	기능추가 ▾	중요함 ▾	문제 제기 ▾	모두
상태:	상태 숨김:					우선 순위:
모두	다음 (이상)					모두
갯수:	상태 보기:	Sticky 이슈 보기:	변경된(시간):	날짜 필터 사용:	관계:	
50	공개 ▾	예	6	아니오	모두	
플랫폼:	운영체제:	버전:	태그:			
모두	모두	모두				
메모를 단 사람:	모두	정렬 기준:	업데이트된 내림차순			
일치 방식:	모든 조건					
🔍 검색 <input type="text"/> <input type="button" value="필터"/> [고급 필터] [고유 링크 만들기] <input type="button" value="필터 초기화"/> <input type="button" value="현재 필터 저장"/>						

Mantis

Issue(cont.)

- 이슈를 선택하면 올라와 있는 이슈를 자세히 볼 수 있다. 원하면 모니터링 하여 상태를 볼 수 있다.



ID	프로젝트	분류	상태 보기	보고 날짜	최종 경신
0000005	MyBuildingServer	기능추가	공개	2013-04-18 16:41	2013-04-18 16:41
보고자	papimomi				
이슈 할당	AuraWalker				
우선 순위	보통	중요도(심각성)	기능 개선	재발생 가능성	N/A
상태	해결된 이슈		해결 상태	개설	
플랫폼	운영체제			버전	
제품 번호	1.0.x				
목표 버전	1.1.x		해결된 버전		
요약	0000005: 건물 스프라이트 변경				
설명	건물에 따라서 이미지를 만들어가지고 랜덤으로 보여질 수 있도록 기능을 추가				
태그	붙인 태그가 없습니다.				
태그 붙이기	("," 사용자가 분리) <input type="text"/> <input type="button" value="기준 태그 ->"/> <input type="button" value="붙이기"/>				

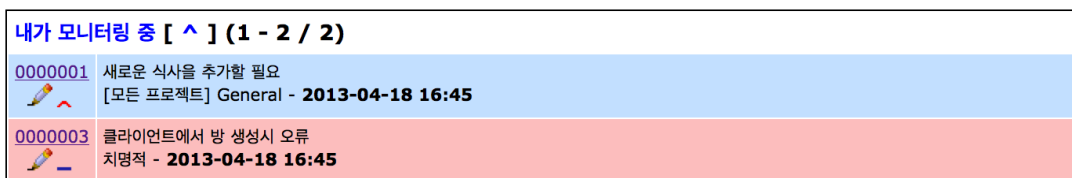
원부 파일

관계

새로운 관계

파일 업로드

- 모니터링을 하면 다음과 같이 '내 페이지'에서 요약된 항목들 중 '내가 모니터링 중' 부분에서 내가 모니터링을 선택한 이슈들을 보여준다.



내가 모니터링 중 [^] (1 - 2 / 2)

0000001	새로운 식사를 추가할 필요 [모든 프로젝트] General - 2013-04-18 16:45
0000003	클라이언트에서 방 생성시 오류 치명적 - 2013-04-18 16:45

Mantis

Issue(cont.)

- 나에게 할당 된 이슈가 있다면 첫 페이지에 할당 된 이슈를 보여주고, 선택을 한다면 그 이슈들을 출력해준다.

나에게 할당되고 진행중인 이슈: 2

이슈 보기 (1 - 2 / 2) [보고서 출력] [CSV 내보내기] [Excel 내보내기]

		P	ID	#	이	분류	중요도(심각성)	상태	업데이트됨 ▼	요약
<input type="checkbox"/>		^	0000004			매우 급함	장애	할당된 이슈 (papimomi)	2013-04-18	건물이 세워지지 않는 오류
<input type="checkbox"/>		-	0000002			기능추가	보통	할당된 이슈 (papimomi)	2013-04-18	빌딩을 모두 파괴

모두 선택 이동 확인

Mantis

Roadmap

- 이슈 보고할 때 버전 선택에서 목표 버전을 설정하면 해당 버전까지 이슈가 등록되고 버전 완료에 대한 이슈를 보여준다.

MyBuildingServer - 로드맵

[MyBuildingServer - 1.1.x](#) (릴리스 예정 2013-04-17) [[이슈 보기](#)]

=====



- [0000003](#): **[치명적]** 클라이언트에서 방 생성시 오류 - 신규.
- [0000005](#): **[기능추가]** 건물 스프라이트 변경 ([AuraWalker](#)) - 할당된 이슈.
- [0000002](#): **[기능추가]** 빌딩을 모두 파괴 ([papimomi](#)) - 해결된 이슈.

전체 3개 중 1개 이슈를 해결했습니다. 진행 (33%).

Mantis

with Subversion

- Mantis와 Subversion을 연동하여 이슈 메모에 커밋 내용을 등록할 수 있음
- 해당 이슈를 해결할 수 있음
- 해당 이슈에 대해 메모를 할 수 있음

☐ 메모	
(0000001) papimomi (매니저) [비공개] 2013-04-18 17:39	Test fixed issue 0000004 by papimomi
<input type="button" value="편집"/> <input type="button" value="삭제"/> <input type="button" value="공개로 하기"/>	
(0000002) papimomi (매니저) [비공개] 2013-04-18 17:45	Test issue 0000004 by papimomi
<input type="button" value="편집"/> <input type="button" value="삭제"/> <input type="button" value="공개로 하기"/>	

Mantis

with Subversion(cont.)

- 특정 이슈에 다음과 같은 커밋 메시지를 작성해주면 노트로 올라간다.
(작성 가능) issue #번호 (커밋 내용)

```
maegdoli:trunk papimomi$ svn commit -m "issue #2 서버 부분에서 클라이언트 연결  
부분 수정 "  
전송 중      src/BuildingServerListener.java  
파일 데이터 전송 중 .  
커밋된 리비전 13.
```

[\(0000006\)](#)

papimomi (매니저) [비공개]

2013-04-18 20:40

편집

삭제

공개로 하기

issue [0000002](#) 서버 부분에서 클라이언트 연결 부분 수정

Chapter 2

Subversion

Subversion

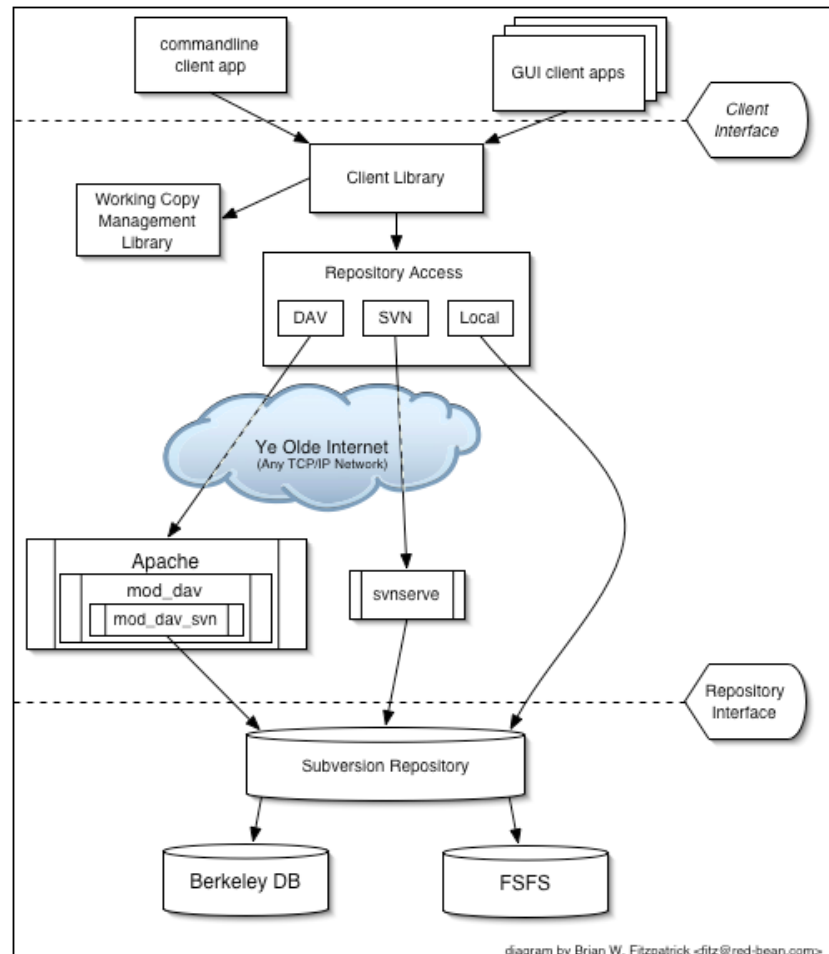
What is SVN?

- 소스 코드의 버전 관리 시스템
- 과거의 어떤 시점으로 돌아갈 수 있다.
- 파일을 누가 수정했는지 추적할 수 있다.
- 파일의 특정 부분이 왜 그렇게 쓰여 졌는지 의미를 추적할 수 있다.
- Branch로 프로젝트에 영향을 최소화 하며 새로운 부분을 개발할 수 있다.

Subversion

What is SVN?

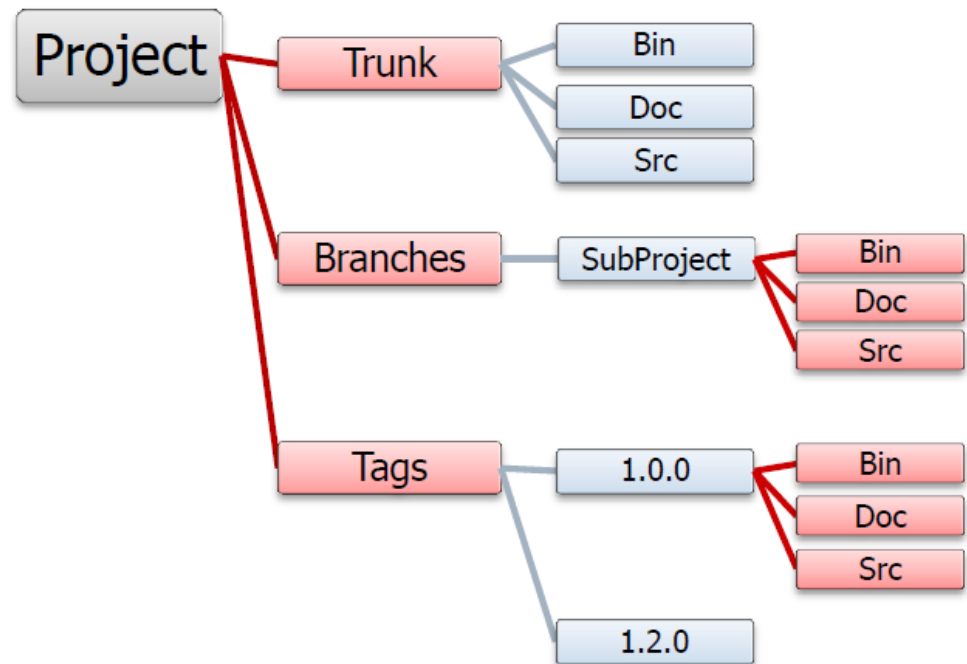
- 서버와 클라이언트로 구성
- 서버 내부의 프로젝트마다 Repository 존재
- 각각의 클라이언트는 하나의 Repository에 접근하여 소스를 읽거나 갱신



Subversion

What is SVN Repository?

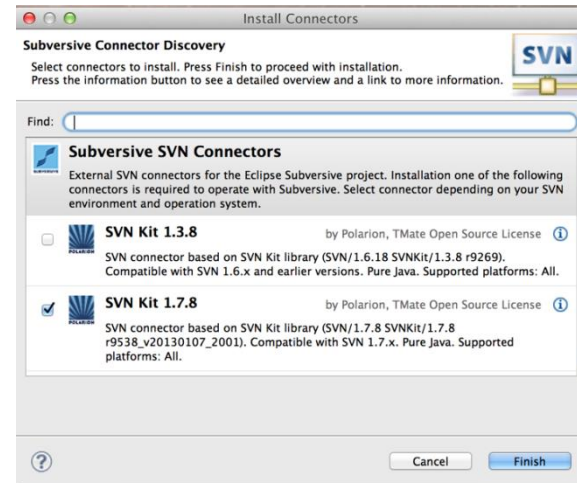
- Trunk
프로젝트의 중심이 되는 폴더
실제 개발은 이곳에서 진행
- Branches
Trunk에서 뺀어 나온 또 하나의
프로젝트로 메인 프로젝트에서
분기하여 따로 개발
- Tags
버전 업이나 특정 시점에서
프로젝트 전체를 복사하여 관리



Subversion

Subversion in Eclipse

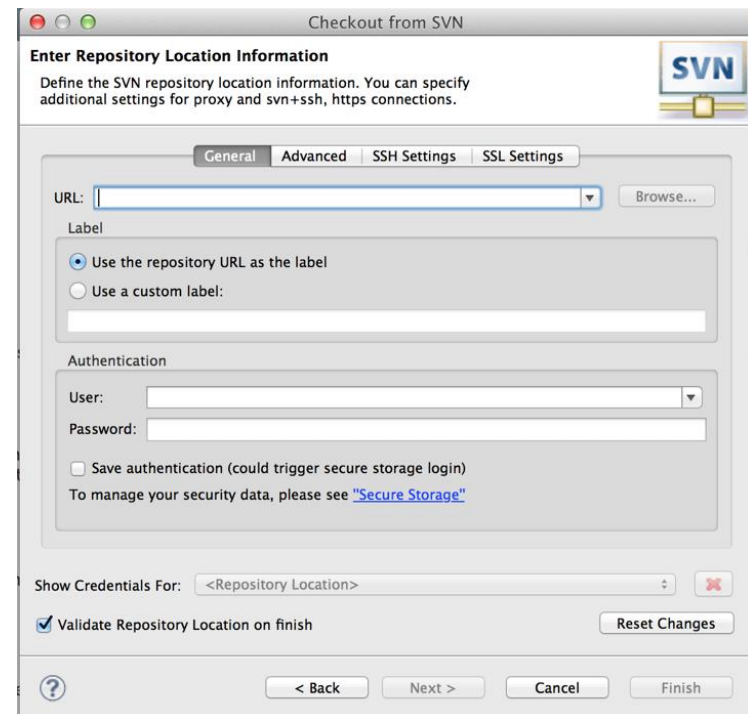
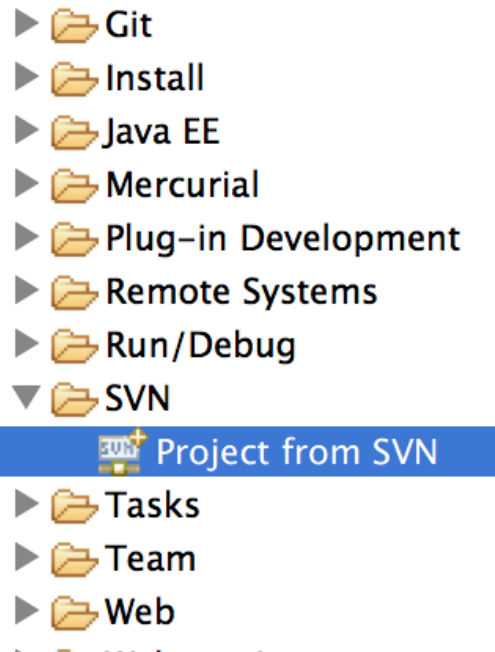
- help→Eclipse Marketplace→svn 검색
- Subversive → SVN Team Provider 선택 후 설치
- eclipse 재시작 후 SVN kit 1.3.8 선택 후 설치



Subversion

Checkout in Eclipse

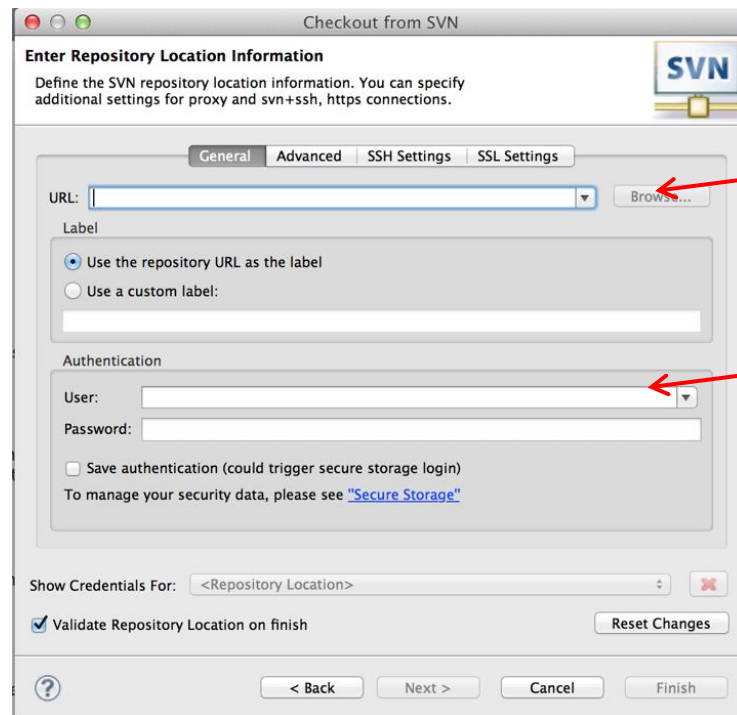
- File -> Import -> SVN -> Project from SVN



Subversion

Checkout in Eclipse(cont.)

- URL에 **svn://dev.viskit.co.kr:23867/2014SV** 입력
- 소프트웨어 검증 팀에게 사용자 계정 및 패스워드를 발급받아 입력



Subversion

Commit / Update

- Project 우클릭 -> Team -> Commit / Update

The image shows a screenshot of an IDE interface. On the left, the Project Explorer shows a project structure. A context menu is open over the project, with 'Team' selected. The 'Team' submenu is also open, and 'Commit...' is highlighted. A red arrow points from the 'Commit...' option to a 'Commit' dialog box on the right. The dialog box has a title bar 'Commit' and an SVN logo. It contains a text area for 'Enter a commit comment' with 'Commit Test' entered. Below the text area is a dropdown for 'Choose a previously entered comment or template'. There is a checkbox for 'Keep Locks' and a 'Paste selected names' button. A table lists resources with their content status:

Resource	Content	Properties
<input checked="" type="checkbox"/> BuildingServer/build.xml	Modified	
<input checked="" type="checkbox"/> BuildingServer/build2.xml	New	
<input checked="" type="checkbox"/> BuildingServer/build3.xml	New	

At the bottom of the dialog, there are buttons for 'Select All', 'Clear Selection', and 'Clear All', and a status indicator 'Selected: 3 of 3'. 'OK' and 'Cancel' buttons are at the bottom right.

Subversion

Commit / Update(cont.)

- **Update**
저장소의 소스폴더와 비교하여 변경된 부분을 검사하고 최신 버전으로 동기화를 맞춤

- **Commit**
소스를 변경한 후 저장소에 반영
Commit할 때마다 버전이 증가하여 변경 사항이 기록

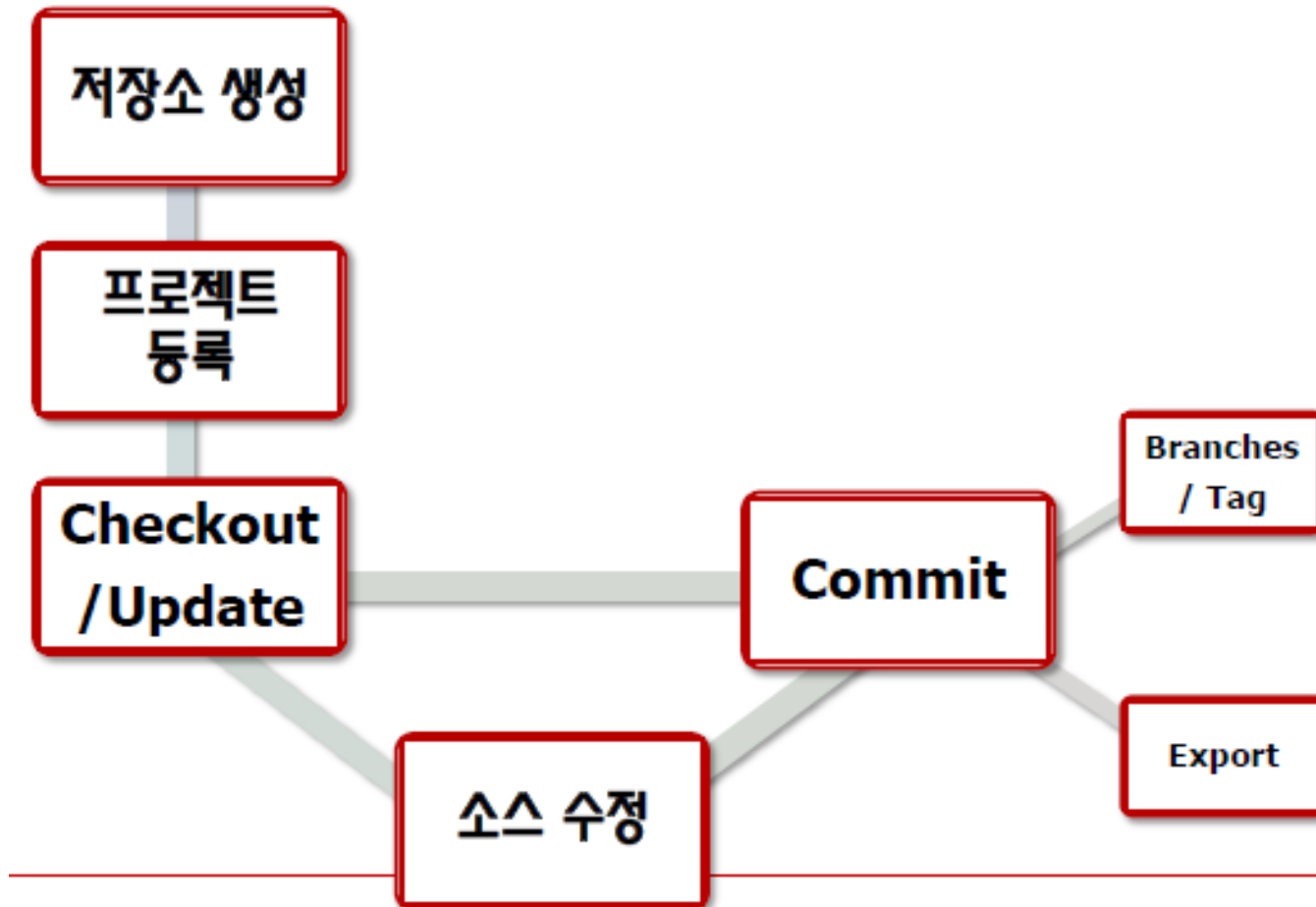
Subversion

Commit / Update(cont.)

- 먼저 작업을 끝낸 개발자가 Commit하게 되면, 이후에 Commit하는 작업자들은 Commit **실패**
- Commit 실패한 개발자들이 Update하게 되면 최신버전의 소스코드와 자신의 소스코드를 **Merge**
- Update하게 되면 그 **소스파일의 버전**이 바뀜

Subversion

SVN 개발 흐름도



Chapter 3

JUnit

JUnit

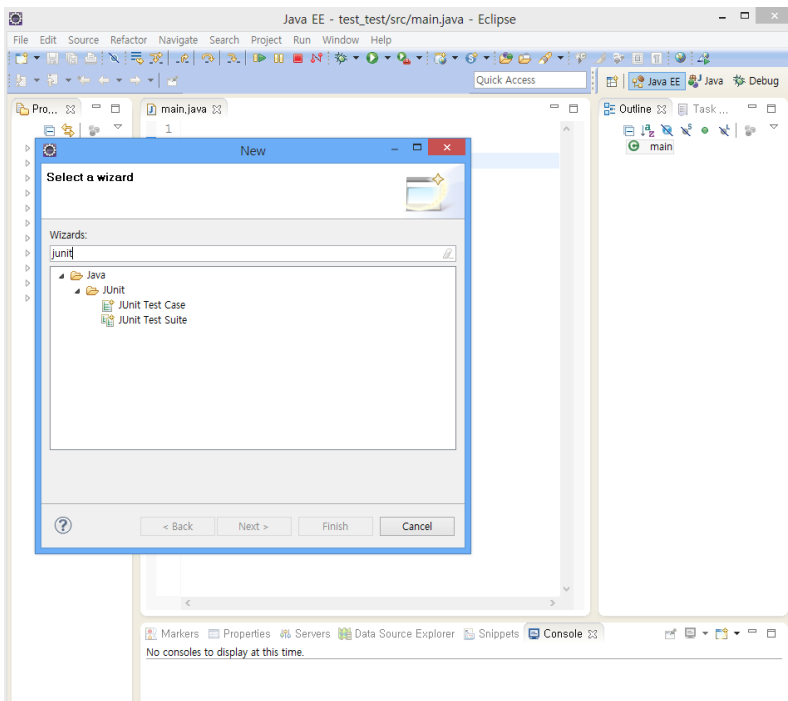
What is JUnit?

- Java 프로그래밍을 할 때 Unit Test를 쉽게 해주는 Framework이다.
- Unit 단위는 하나의 메소드로 간주된다.
- 결과값과 기대값이 같은지를 판단하여 성공과 실패를 결정한다.

JUnit

Install

Eclipse 2.1 버전 이상에서는 JUnit을 기본적으로 사용.



메뉴에서 File->New->Other 에서 Junit을 입력하여 확인 할 수 있다.

JUnit

Annotation

@Test	Unit test 대상 method를 정의
@Test(timeout)	테스트 시간을 예측할 때 사용 시간보다 길게 진행될 시 Fail
@Test(expected)	예외를 지정할 때 사용 예외가 발생하지 않을 시 Fail
@Ignore	테스트 하지 않을 method 앞에 작성. 이후에 오는 테스트를 무시한다.
@After @Before	각 단위 테스트 method의 실행 앞, 뒤에서 초기화 및 자원 정리
@AfterClass @BeforeClass	각 단위 테스트 class 수행 전, 후 에 초기화 및 자원 정리

JUnit

Annotation(Cont.)

@RunWith	사용자가 지정한 러너를 통해 특정 클래스를 실행
@SuiteClasses	테스트 하고자 하는 다수의 클래스를 지정
@Parameters	다수의 parameter값을 테스트 하려고 할 때 자동으로 테스트를 실행

JUnit

Method

assertEquals (타입 expected, 타입 result)	예상값과 결과값을 매개변수로 지정하고 두 값이 같을 경우에 성공. 타입에는 int, long 등의 기본 자료형과 String, Object가 사용 가능
assertEquals (String Message, 타입 expected, 타입 result)	위에 assertEquals에서 예상 값과 결과 값이 같지 않을 경우 첫 번째 매개변수인 Message 값을 리턴
assertTrue (boolean condition)	condition이 true이면 성공, false이면 실패로 처리한다
assertFalse (boolean condition)	condition이 true이면 실패, false이면 성공으로 처리한다

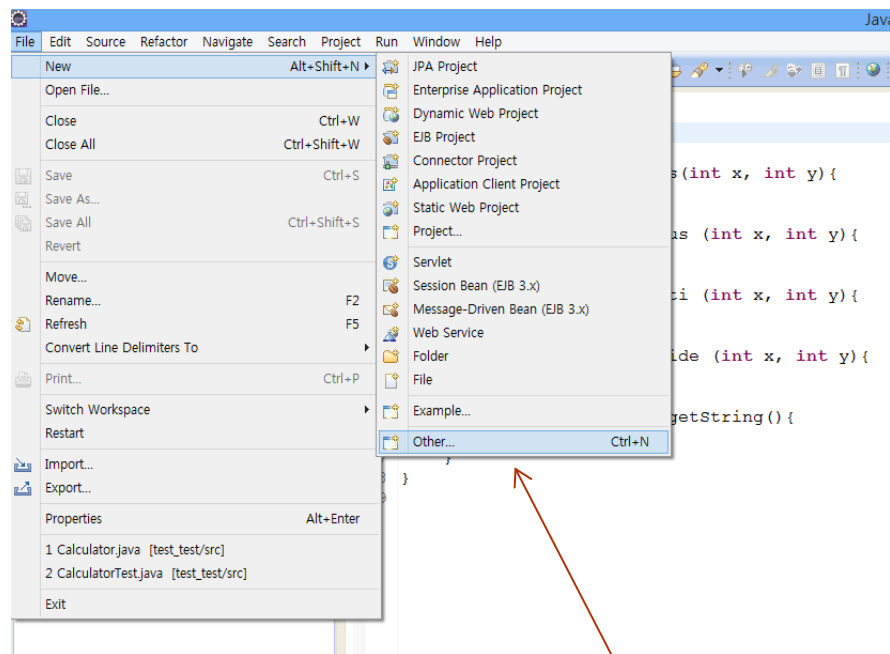
JUnit

Method(Cont.)

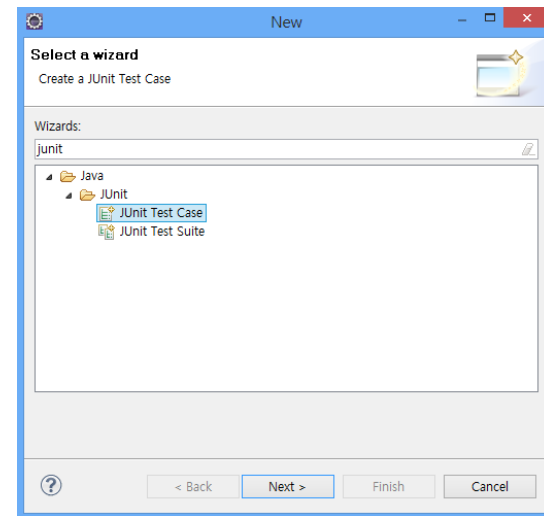
assertNotNull (Object obj)	객체가 null이 아니면 성공
assertNull (Object obj)	객체가 null이면 성공
assertSame (Object expected, Object result)	객체가 동일하면 성공, 다르면 실패
assertNotSame (Object expected, Object result)	객체가 다르면 성공, 같으면 실패
Fail() 관련 method	테스트가 실패일 경우를 쉽게 처리하는 것과 관련된 method

JUnit

Example 1



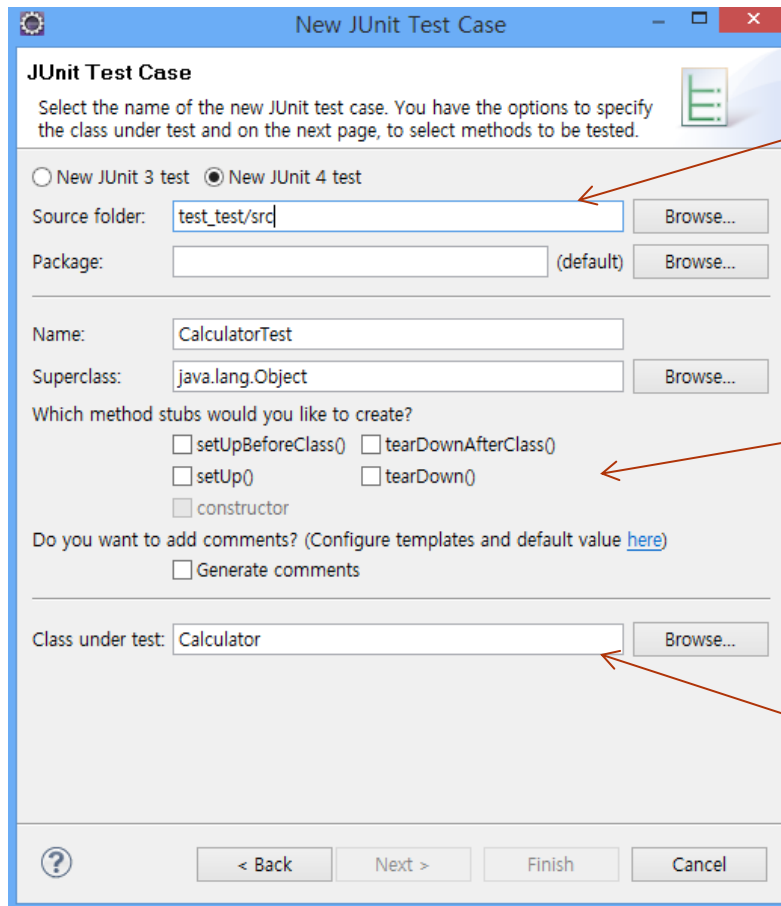
File – New – Other..을 Click



JUnit 검색 – JUnit Test Case 선택

JUnit

Example 1 (Cont.)



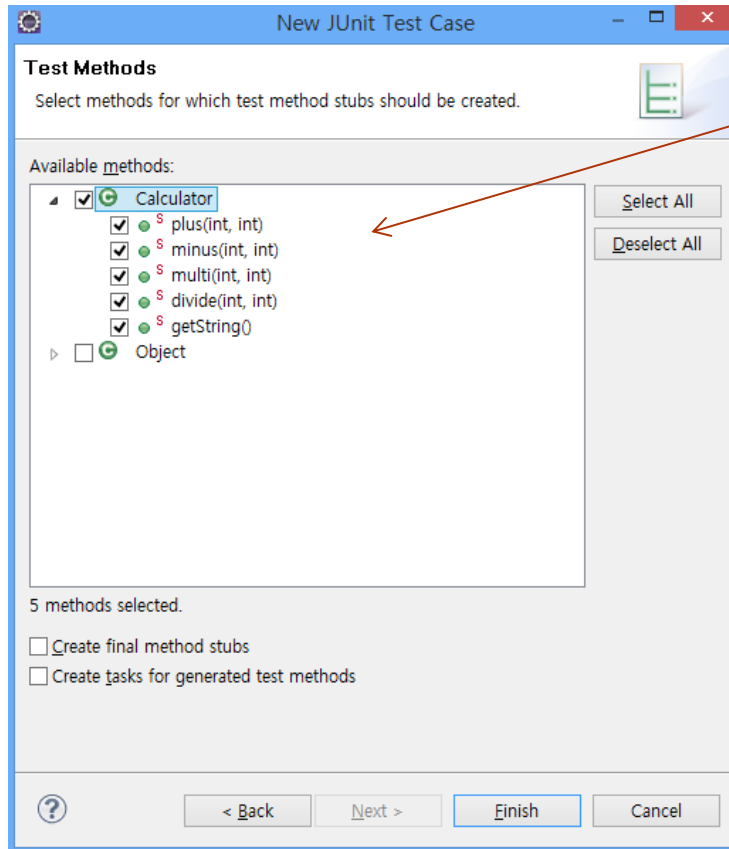
Test 할 Source code 경
로

setUp() : 테스트 대상 클래스
의 객체를 생성하거나 네트워크
연결, DB 연결 작업 등을 수행.
tearDown() : setUp과 정 반대
의 기능. 객체의 제거, 네트워크
종료, DB연결 종료 등을 수행
...beforeClass() : 클래스의 시
작 혹은 끝나고 실행되는
method

Test 할
Case

JUnit

Example 1 (Cont.)



Method 선택

JUnit

Example 1 (Cont.)

```
public class MainTest
{
    @Before
    public void setUp() throws Exception
    {
        //set up before start test
    }

    @After
    public void tearDown() throws Exception
    {
        //tear down after test
    }

    @Test
    public void testMain()
    {
        fail("Not yet implemented");
    } //test main

    @Test
    public void testTenTobin()
    {
        fail("Not yet implemented");
    } //test ten to bin

    @Test
    public void testDivisor()
    {
        fail("Not yet implemented");
    } //test divisor
} //main test
```



```
Main test;

@Before
public void setUp() throws Exception
{
    test = new Main();
    //set up before start test
}

@After
public void tearDown() throws Exception
{
    //tear down after test
}

@Test
public void testTenTobin()
{
    int user=21;

    assertEquals(test.tenTobin(user),10101);

    //fail("Not yet implemented");
} //test ten to bin

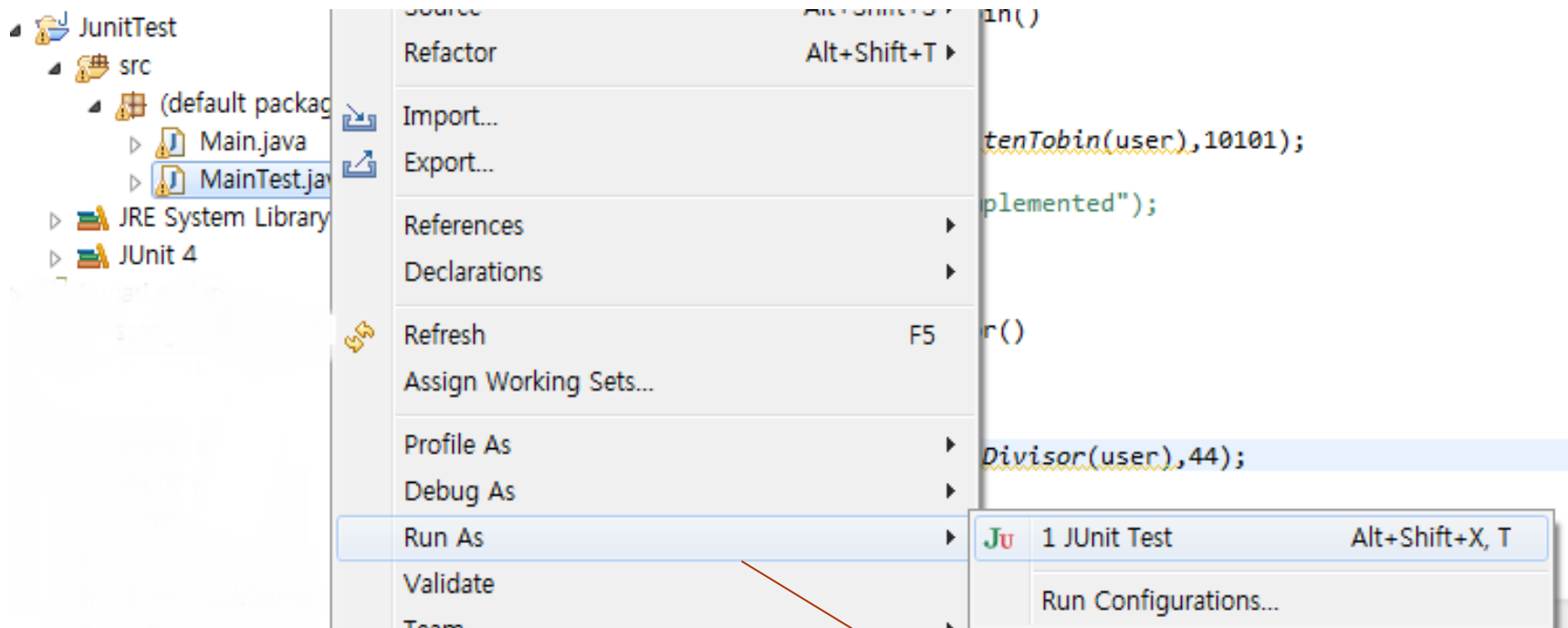
@Test
public void testDivisor()
{
    int user=123;

    assertEquals(test.Divisor(user),44);

    //fail("Not yet implemented");
} //test divisor
```

JUnit

Example 1 (Cont.)

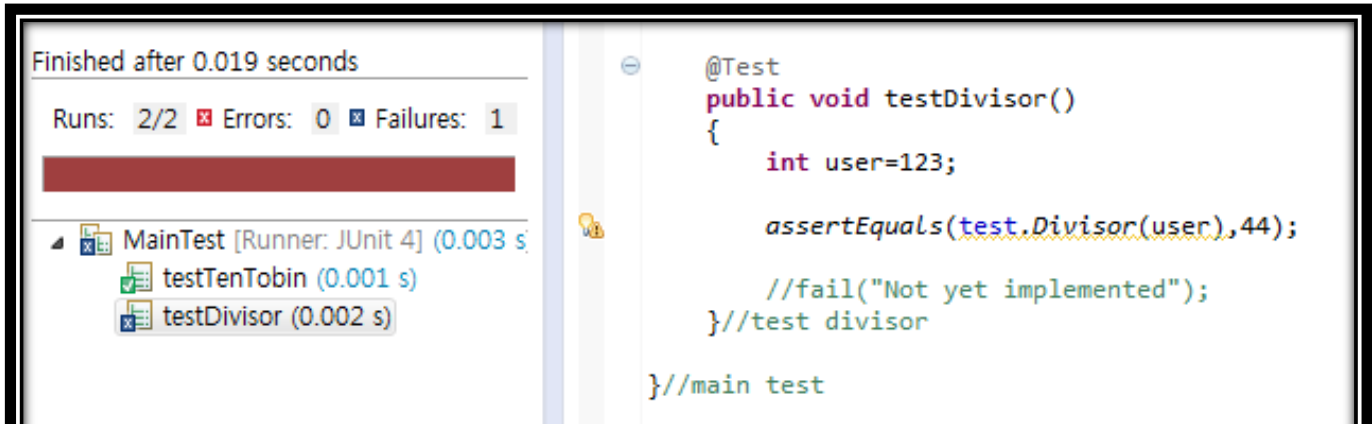


Run as -> Junit Test

JUnit

Example1 (Cont.)

실패 시 화면



Finished after 0.019 seconds

Runs: 2/2 ✖ Errors: 0 ⓧ Failures: 1

MainTest [Runner: JUnit 4] (0.003 s)

- testTenTobin (0.001 s)
- testDivisor (0.002 s)

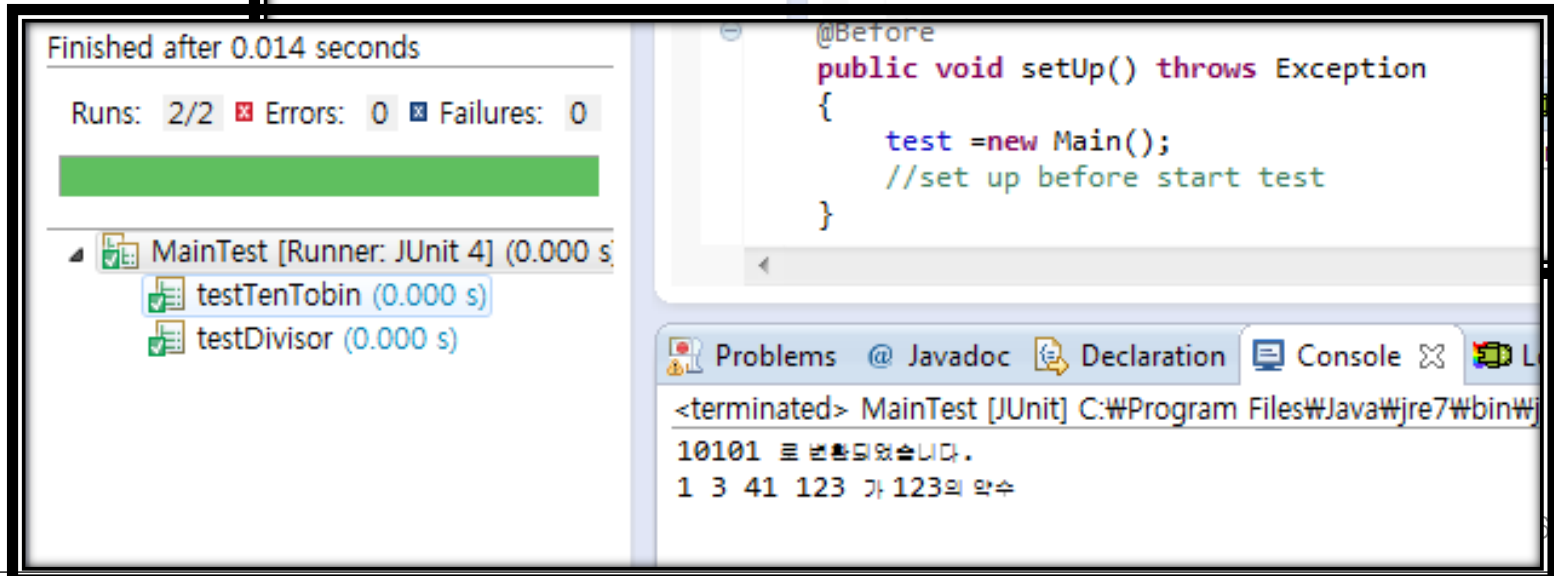
```
@Test
public void testDivisor()
{
    int user=123;

    assertEquals(test.Divisor(user),44);

    //fail("Not yet implemented");
} //test divisor

} //main test
```

성공 시 화면



Finished after 0.014 seconds

Runs: 2/2 ✖ Errors: 0 ⓧ Failures: 0

MainTest [Runner: JUnit 4] (0.000 s)

- testTenTobin (0.000 s)
- testDivisor (0.000 s)

```
@Before
public void setUp() throws Exception
{
    test =new Main();
    //set up before start test
}
```

Problems @ Javadoc Declaration Console

<terminated> MainTest [JUnit] C:\Program Files\Java\jre7\bin\W
10101 크 변경되었습니다.
1 3 41 123 가 123의 약수

JUnit

Example2

```
import static org.junit.Assert.*;
import org.junit.Test;
import org.junit.BeforeClass;
import org.junit.Before;
import org.junit.AfterClass;
import org.junit.After;

public class CalculatorTest {

    static private int i, j;

    @BeforeClass
    public static void ExeBeforeTest() {
        i = 3;
        j = 2;
        System.out.println("Execute @BeforeClass");
    }

    @Before
    public void testPlus() {
        int result = Calculator.plus(1, 2);
        assertEquals(result, 3);
        System.out.println("Execute @Before");
    }

    @Test
    public void TestMinus() {
        int k = i - j;
        assertEquals(1, k);
        System.out.println("Execute @TestMinus");
    }
}
```

BeforeClass는 Test 내에서 가장 먼저 1회 실행된다.

Before는 Unit Test 앞부분에서 Test마다 1회 실행된다.

JUnit

Example2(Cont.)

```
@Test
public void TestPrint1() {
    System.out.println("Execute @Test_1");
}

@Test
public void TestPrint2() {
    System.out.println("Execute @Test_2");
}

@After
public void TestMulti() {
    int k = i * j;
    assertEquals(6, k);
    System.out.println("Execute @After");
}

@AfterClass
public static void ExeAfterTest() {
    System.out.println("Execute @AfterClass");
}
}
```

After는 Unit Test 뒷부분에서 Test마다 1회 실행된다.

AfterClass는 Test 내에서 가장 마지막에 1회 실행된다.

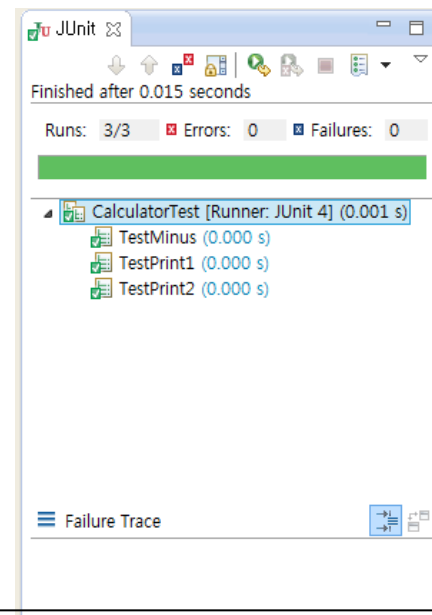
JUnit

Example2(Cont.)

Console Markers Properties

```
<terminated> CalculatorTest [JUnit] C:#Prog  
Execute @BeforeClass  
Execute @Before  
Execute @TestMinus  
Execute @After  
Execute @Before  
Execute @Test_1  
Execute @After  
Execute @Before  
Execute @Test_2  
Execute @After  
Execute @AfterClass
```

각각의 Annotation에 대한 실행 수와 실행 순서를 Print문을 통해 확인 할 수 있다.



@Test Annotation을 제외한 다른 Annotation의 경우 Testing은 이뤄지지 않는다.