

SOFTWARE VERIFICATION #2

3조 송지연, 윤상혁, 장서연



01

Redmine

Redmine

Mantis

Install

02

Github

Github

How to use

03

ReqView

JFeature

ReqView

Install

04

CTIP

CTIP



01

Redmine

Redmine

Mantis

Install

01

Redmine

Redmine

Mantis

Install



- ▶ 웹 기반의 프로젝트 관리 기능을 제공하는 웹 애플리케이션
- ▶ Ruby로 제작되어 Ruby도 설치해야 한다
- ▶ Windows 환경에서 쉽게 설치할 수 있는 쉬운 과정으로 전개

01

Mantis

Redmine

Mantis

Install



- ▶ 웹 기반의 버그 및 이슈 관리 시스템
- ▶ 이슈 상태 변경 알림 메일
- ▶ 사전 설치 도구: PHP, MySQL, Apache2

Software Engineering [Jenkins] x Redmine Cloud Hosting, Redi x +

← → ↻ 🏠 <https://bitnami.com/stack/redmine> 📄 ⋮ 📧 ☆ ⏴ ⏵ ⏶ ⏷ ⏸ ⏹ ⏺ ⏻ ⏼ ⏽ ⏾ ⏿

bitnami Products Solutions Partners Company News Support [Sign In](#)

Home > Applications > Bug Tracking > Redmine

Overview Installers Virtual Machines Cloud Containers

Redmine [Follow](#)
www.redmine.org | Open Source

The Bitnami Redmine Stack provides a one-click install solution for Redmine. Download installers and virtual machines, or run your own Redmine server in the cloud.

Redmine is a very popular Rails-based open source bug tracker to help you manage issues and tasks for multiple projects. It is extremely flexible, features a built-in wiki, time tracking, custom fields, role-based access, SCM integration (including git), and support for multiple projects. Redmine has many of the same features you'd find in an enterprise bug tracking system, but of course it's open source and completely free software. Redmine has a clean interface that makes it easy to define your first project and get started.

More on Redmine

Redmine will also integrate with a version control tool so that development and project management workflows can be tightly connected with each other. The Bitnami Redmine Stack comes with Subversion already installed and configured.

Key features of Redmine include

- Add users and roles.
- Add and track issues (or have your team do it).
- Plan and manage your project with Redmine's built-in Gantt charts and calendar.
- Use the project wiki and document manager to store project documents.
- Configure notifications to stay updated on issue status and changes.
- Define workflows to streamline collaboration with your team.

DEMO IN BROWSER

LAUNCH IN THE CLOUD

CONTAINER DEPLOYMENT

Redmine Docker Container

LOCAL INSTALL

DOWNLOAD REDMINE INSTALLER

[bitnami-redmine-3.4.4-3-windows-in...](#)
Version 3.4.4

OTHER OPTIONS

All Redmine Installers

Redmine Virtual Machines

[Help me choose](#)

<https://bitnami.com/redirect/to/181960/bitnami-redmine-3.4.4-3-windows-installer.exe> the Bitnami Redmine Stack?

01

Redmine

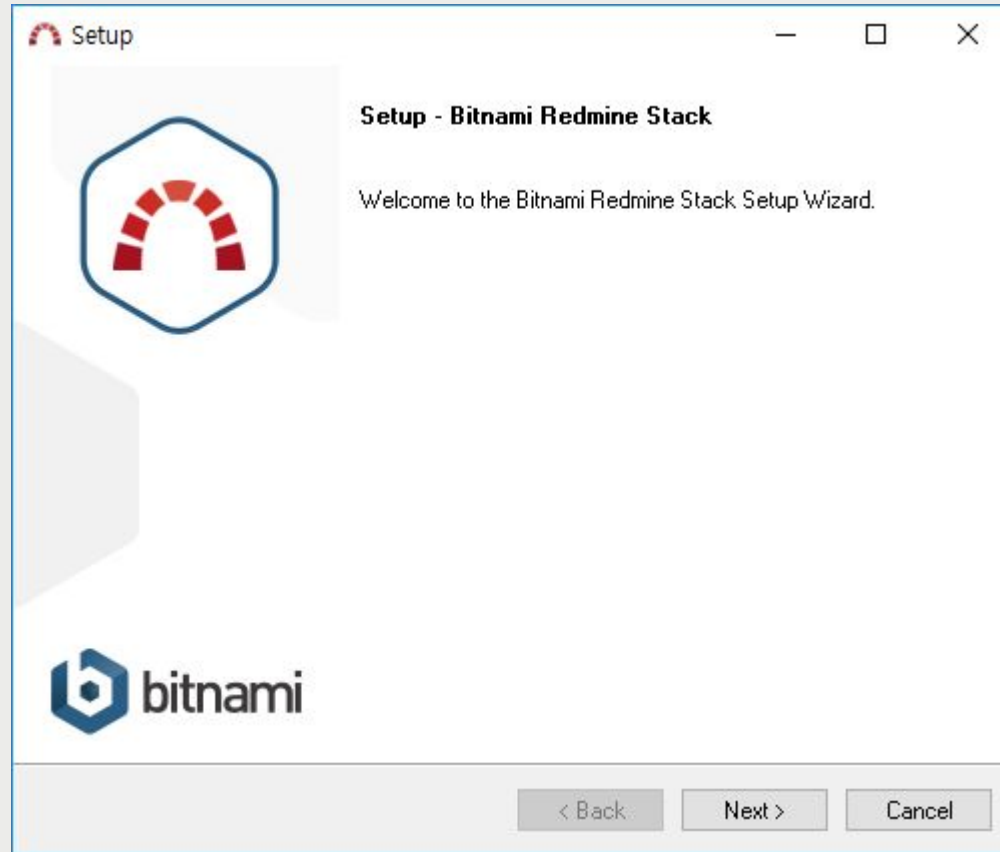
Redmine

Mantis

Install

| 설치

설치 파일을 열어줍니다.



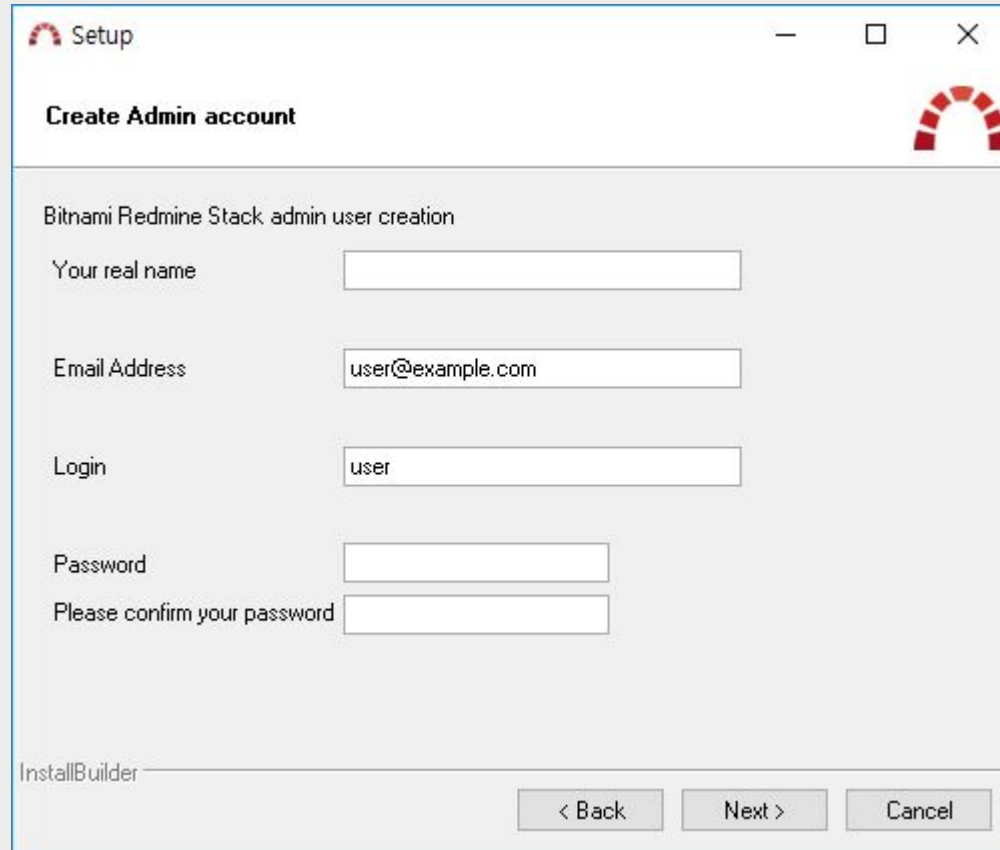
Redmine

Mantis

Install

| 설치

Admin 정보를 작성합니다.



The screenshot shows a Windows-style window titled "Setup" with a Bitnami logo in the top-left corner. The window content is titled "Create Admin account" and features a Bitnami logo in the top-right corner. Below the title, the text "Bitnami Redmine Stack admin user creation" is displayed. The form contains five input fields: "Your real name" (empty), "Email Address" (containing "user@example.com"), "Login" (containing "user"), "Password" (empty), and "Please confirm your password" (empty). At the bottom of the window, the text "InstallBuilder" is visible on the left, and three buttons are on the right: "< Back", "Next >", and "Cancel".

Setup

Create Admin account

Bitnami Redmine Stack admin user creation

Your real name

Email Address

Login

Password

Please confirm your password

InstallBuilder

< Back Next > Cancel

01

Redmine

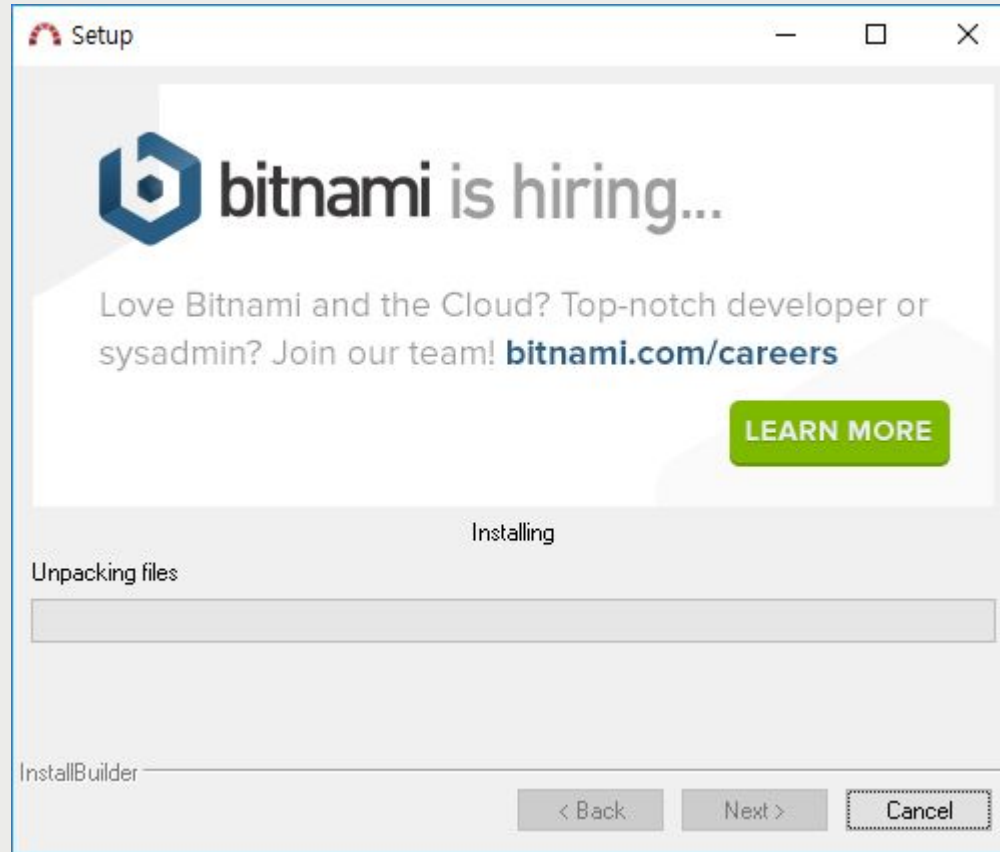
Redmine

Mantis

Install

| 설치

설치가 진행됩니다.



01 Redmine

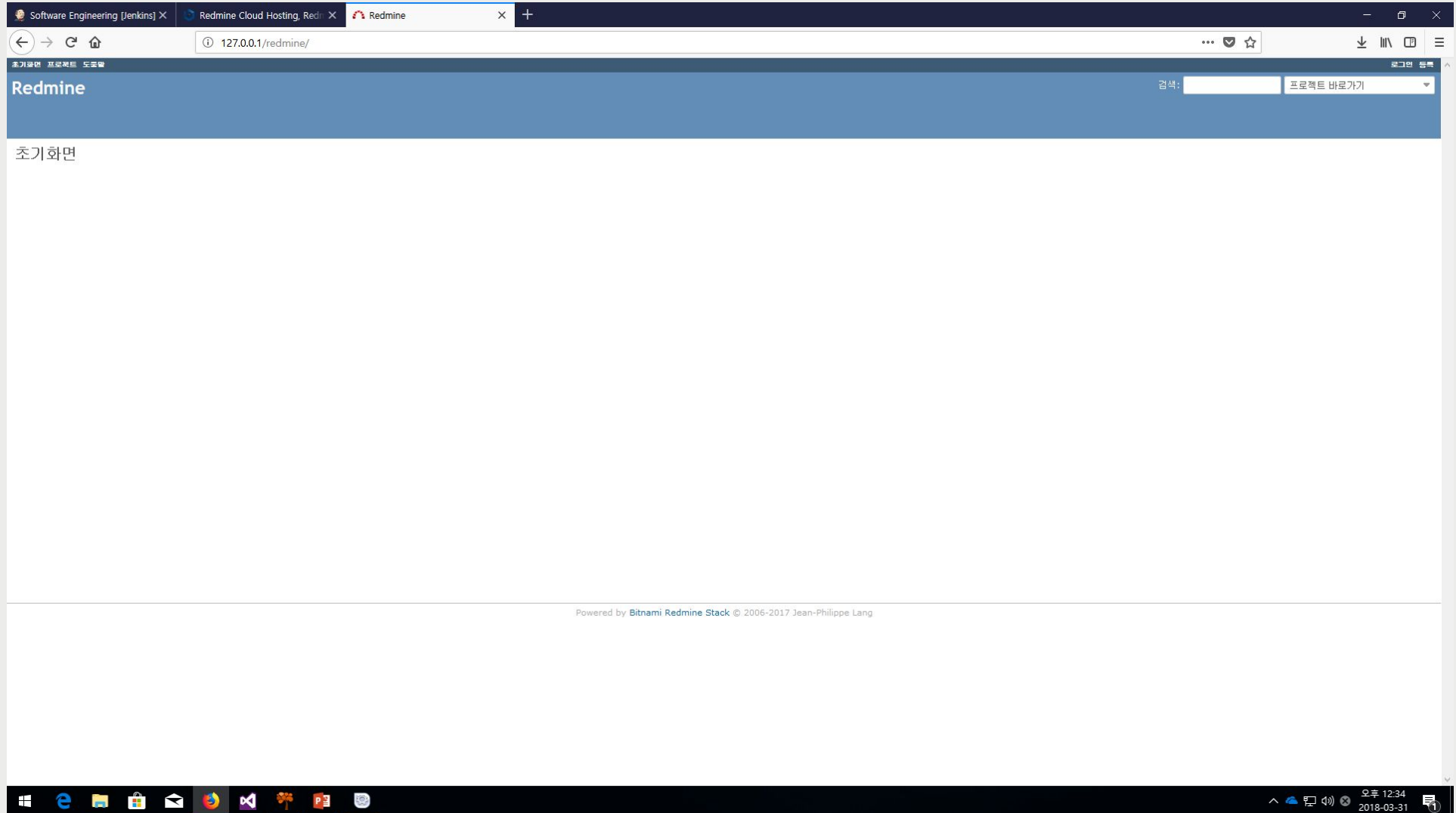
Redmine

Mantis

Install

| 실행

설치가 완료되면 Redmine 관리자 페이지가 열립니다.



01 Redmine

Redmine

Mantis

Install

| 실행

설정된 Admin 정보로 로그인합니다.

The screenshot shows a web browser window with the following details:

- Browser Tabs:** Software Engineering [Jenkins] X, Redmine Cloud Hosting, Redi X, Redmine X
- Address Bar:** 127.0.0.1/redmine/login
- Page Header:** Redmine (left), 검색: (input field), 프로젝트 바로가기 (dropdown menu) (right)
- Form:** A yellow-bordered box containing:
 - 로그인 (Login) label
 - Username input field
 - 비밀번호 (Password) label and 비밀번호 찾기 (Forgot password) link
 - Password input field
 - 로그인 (Login) button
- Footer:** Powered by Bitnami Redmine Stack © 2006-2017 Jean-Philippe Lang
- Taskbar:** Windows taskbar with icons for Start, Edge, Mail, Store, File Explorer, Firefox, VS Code, and others. System tray shows the time as 12:34 PM on 2018-03-31.

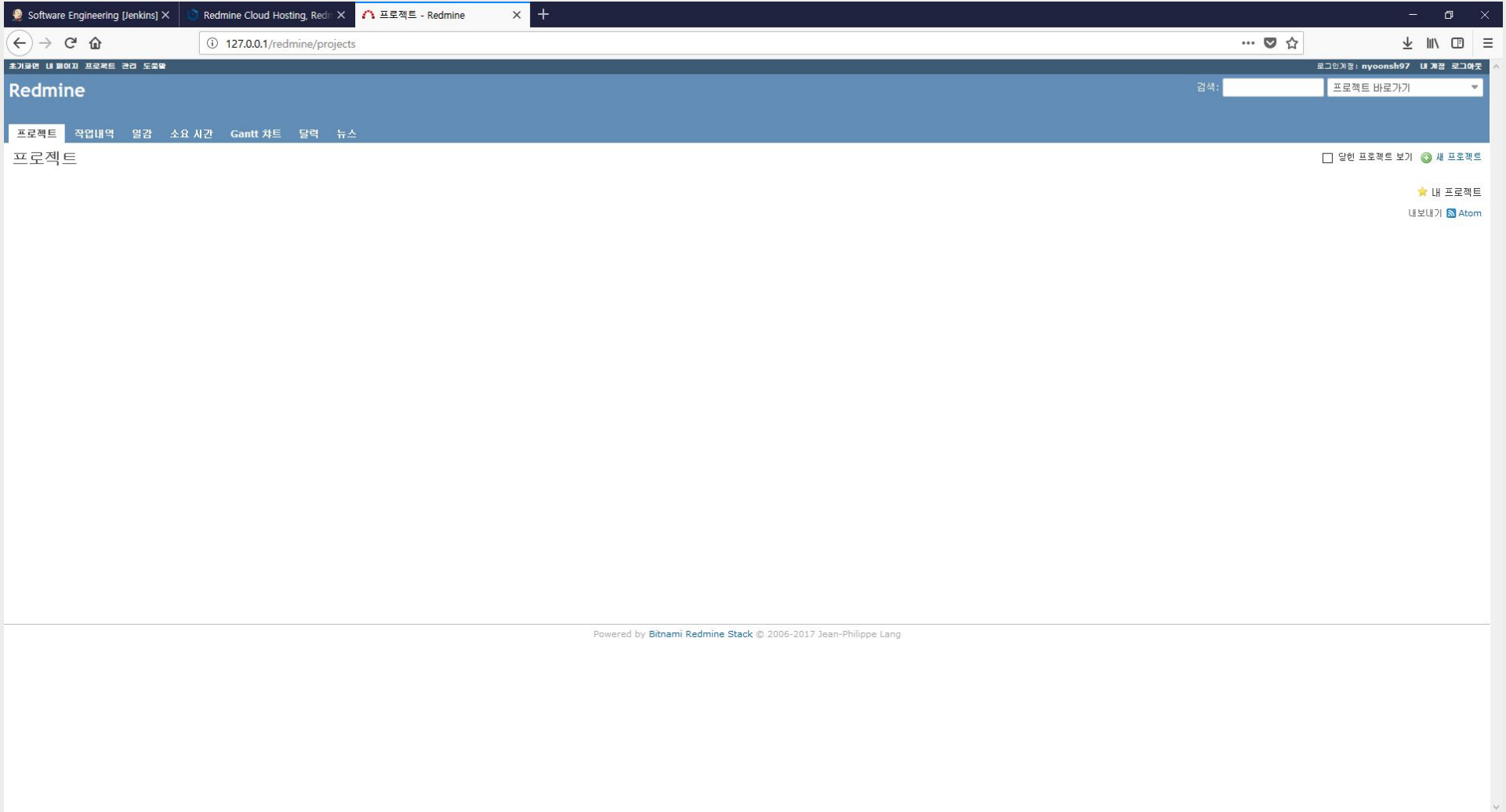
Redmine

Mantis

Install

| 실행

프로젝트 메뉴에서 새 프로젝트를 생성할 수 있습니다.



| 실행

프로젝트 상세 정보를 입력하고 '만들기'로 프로젝트를 생성합니다.

The screenshot shows the Redmine web interface for creating a new project. The browser address bar shows the URL `127.0.0.1/redmine/projects/new`. The page title is "Redmine" and the user is logged in as "nyoonsh97".

The "새 프로젝트" (New Project) form contains the following fields and options:

- 이름 (Name):** Software Engineering
- 설명 (Description):** Test
- 식별자 (Identifier):** software-engineering
- 홈페이지 (Homepage):** (empty)
- 공개 (Public):**
- 상위 프로젝트 구성원 상속 (Inherit parent project settings):**

Below the form, there are sections for selecting modules and issue types:

- 모듈 (Modules):** Includes checkboxes for 일감관리, 저장소, 시간추적, 계시판, 뉴스, 달력, 문서, Gantt 차트, 파일, and 위키.
- 일감 유형 (Issue Types):** Includes checkboxes for Bug, Feature, and Support.

At the bottom of the form, there are two buttons: "만들기" (Create) and "만들고 계속하기" (Create and Continue).

Footer text: "Powered by Bitnami Redmine Stack © 2006-2017 Jean-Philippe Lang"

| 실행

해당 프로젝트의 할 일을 생성합니다.

The screenshot shows a web browser window displaying the Redmine interface for creating a new issue. The browser tabs include 'Software Engineering [Jenki]', 'Redmine Cloud Hosting, Re...', '새 일감만들기 - Software En...', 'redmine jenkins 연동 - Go...', ''artifacts'의 검색결과 : 네이...', 'Test2/Hello.java at master', 'Custom Requirement Attrib...', and 'redmine - Google 검색'. The address bar shows the URL '127.0.0.1/redmine/projects/software-engineering/issues/new'. The page title is 'Software Engineering' and the search bar contains 'Software Engineering'. The navigation menu includes '개요', '작업내역', '일감', '소요 시간', 'Gantt 차트', '달력', '뉴스', '문서', '위키', '파일', and '설정'. The main heading is '새 일감만들기'. The form fields are: '유형' (Bug), '제목' (String 입력 오류), '설명' (Rich text editor), '상태' (New), '우선순위' (Urgent), '담당자' (empty), '상위 일감' (empty), '시작시간' (2018. 03. 31.), '완료기한' (2018. 04. 04.), '추정시간' (12 시간), and '진척도' (0%). There is a '파일' section with a '찾아보기...' button and a note '선택한 파일이 없습니다. (최대 크기: 5 MB)'. At the bottom, there are buttons for '만들기', '만들고 계속하기', and '미리보기'. The footer text is 'Powered by Bitnami Redmine Stack © 2006-2017 Jean-Philippe Lang'.

| 실행

해당 프로젝트의 할 일을 생성합니다.

Software Engineering [Jenki] × Redmine Cloud Hosting, Re × 일감 - Software Engineering × redmine jenkins 연동 - Goo × 'artifacts' 의 검색결과 : 네이 × Test2/Hello.java at master × Custom Requirement Attrib × redmine - Google 검색 × + - □ ×

127.0.0.1/redmine/projects/software-engineering/issues

Software Engineering

검색: Software Engineering

가요 작업내역 일감 소요 시간 Gantt 차트 달력 뉴스 문서 위키 파일 설정

일감

검색조건

상태 진행중

옵션

적용 지우기 저장

#	유형	상태	우선순위	제목	담당자	변경
1	Bug	New	Urgent	String 입력 오류		2018/03/31 15:04

(1-1/1)

내보내기 Atom | CSV | PDF

모든 일감 보기
요약
가져오기

| 실행

Gantt 차트에서 프로젝트의 일정관리를 할 수 있습니다.

The screenshot shows the Redmine Gantt chart interface for a project named "Software Engineering". The browser address bar shows the URL "127.0.0.1/redmine/projects/software-engineering/issues/gantt". The page title is "Software Engineering". The navigation menu includes "개요", "작업내역", "일감", "소요 시간", "Gantt 차트", "달력", "뉴스", "문서", "위키", "파일", and "설정". The "Gantt 차트" tab is active. The Gantt chart displays a task named "Software Engineering" with a progress bar showing 0% completion. The task is scheduled to start on 2018-03-13 and end on 2018-03-14. The chart is organized by month, with columns for 2018-3, 2018-4, 2018-5, 2018-6, 2018-7, and 2018-8. The current view shows days 10 through 35. The interface includes search filters for status (currently set to "진행중") and options for zooming in and out. A sidebar on the right contains "일감" (Tasks) with options to "모든 일감 보기", "요약", and "가져오기". The footer indicates the page is powered by Bitnami Redmine Stack © 2006-2017 Jean-Philippe Lang.

| 실행

프로젝트의 할 일은 '달력'으로 일정을 확인할 수 있습니다.

Software Engineering

달력

검색조건: 상태 진행중

월 3월 년 2018

일요일	월요일	화요일	수요일	목요일	금요일	토요일
9	25	26	27	28	1	2
10	4	5	6	7	8	10
11	11	12	13	14	15	17
12	18	19	20	21	22	24
13	25	26	27	28	29	30
						31

Bug #1: String 입력 오류

● 오늘 시작하는 업무(task)
 ● 오늘 종료하는 업무(task)
 ● 오늘 시작하고 종료하는 업무(task)

Powered by Bitnami Redmine Stack © 2006-2017 Jean-Philippe Lang



02

GitHub

GitHub

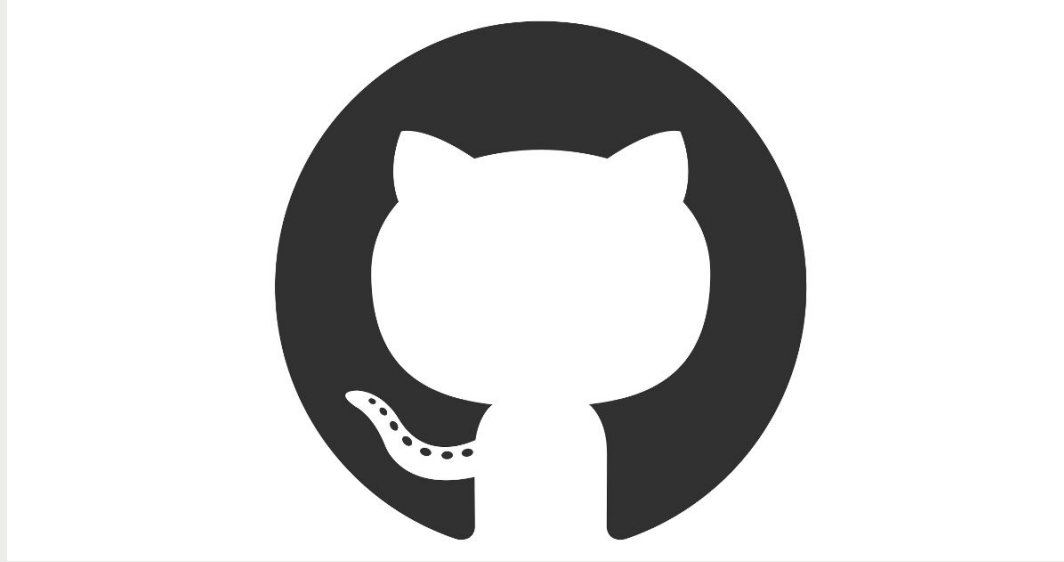
Connect GitHub with eclipse

02

GitHub

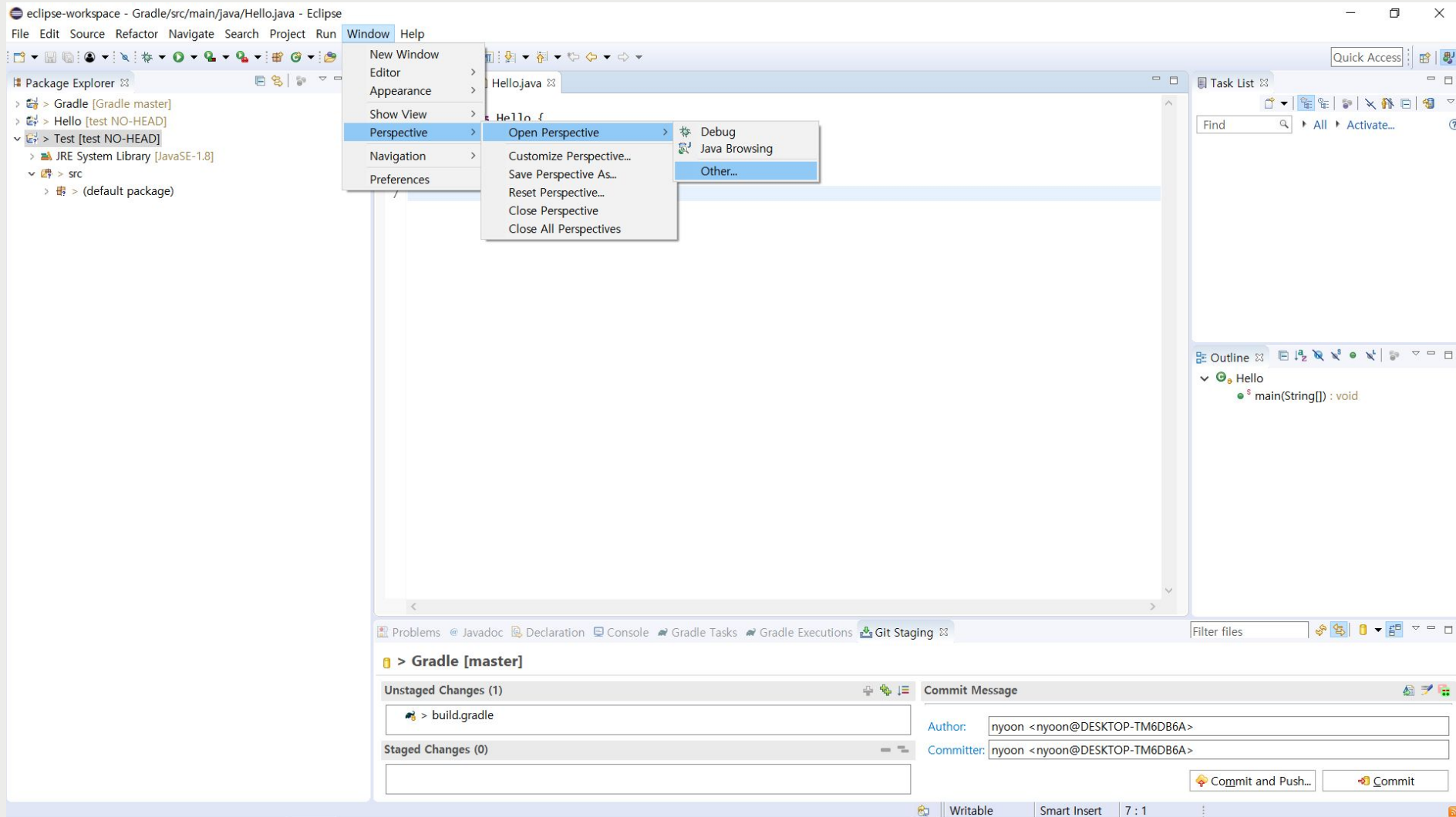
GitHub

Connect GitHub with Eclipse



분산 버전 관리

깃 웹 호스팅 서비스



02

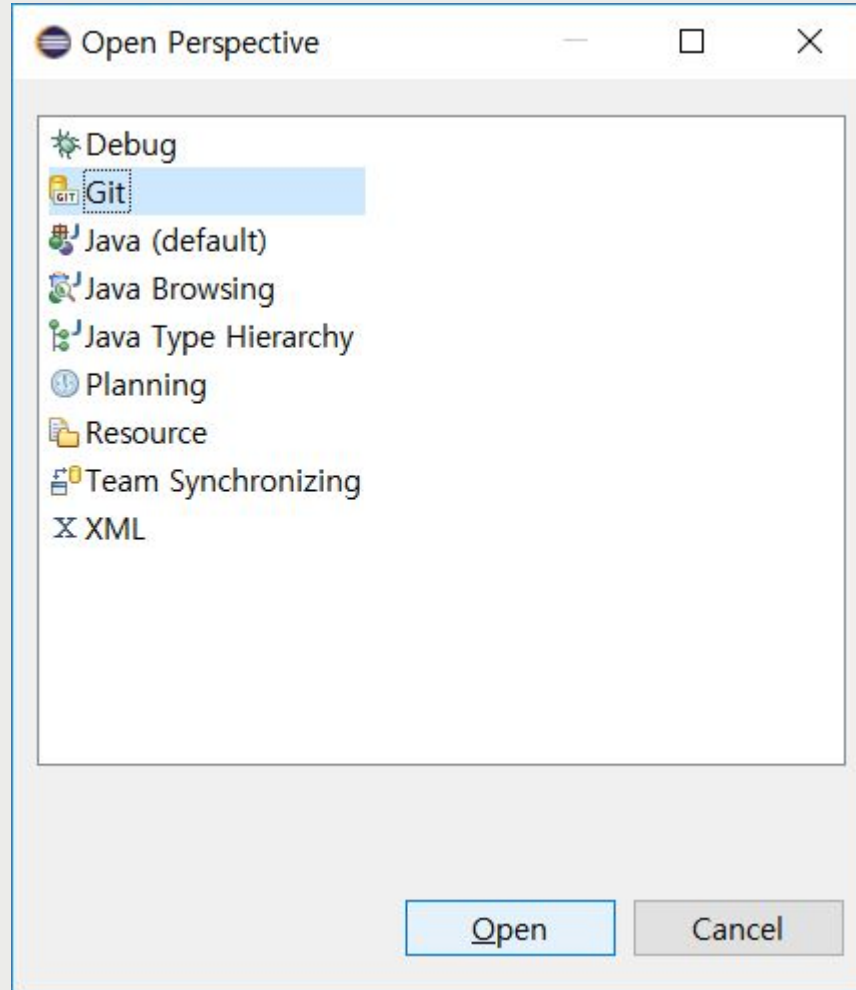
GitHub

GitHub

Connect GitHub with Eclipse

| 연결

Open Perspective에서 Git 선택합니다.



GitHub

Connect GitHub with Eclipse

| 연결

왼쪽에 'Git Repositories'에서 해당하는 메뉴를 선택, Repository를 연결합니다.

The screenshot displays the Eclipse IDE interface. On the left, the 'Git Repositories' view is open, showing a tree structure with folders for 'Gradle', 'test', 'Branches', 'Tags', 'References', 'Remotes', and 'Working Tree'. A red arrow points to the 'Add' icon in the toolbar above this view. The main editor area shows a Java file named 'Hello.java' with the following code:

```
1
2 public class Hello {
3     public static void main(String args[]) {
4         System.out.println("Hello World!");
5     }
6 }
7
```

At the bottom, the 'Properties' view is visible with columns for 'Property' and 'Value'. The status bar at the bottom left indicates '0 items selected'.

| 연결

해당 Git에서 생성된 Git URL과 정보를 입력합니다.

Clone Git Repository

Source Git Repository

Enter the location of the source repository.

Location

URI: Local File...

Host:

Repository path:

Connection

Protocol: ▾

Port:

Authentication

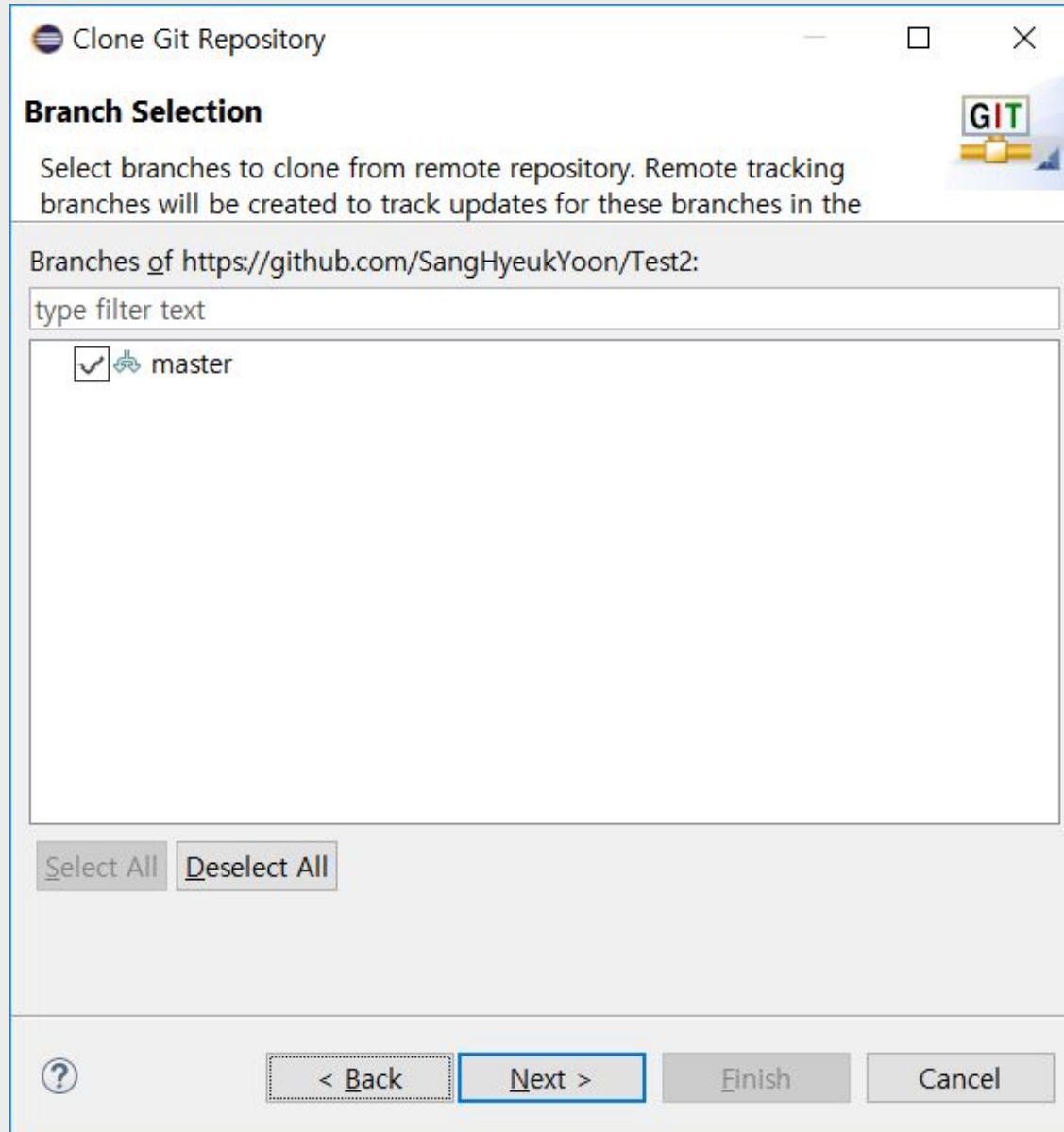
User:

Password:

Store in Secure Store

? < Back Next > Finish Cancel

해당 Repository에서 Clone할 Branch를 선택합니다.



Local 경로를 확인한 뒤, Finish를 선택합니다.

Clone Git Repository

Local Destination

Configure the local storage location for Test2.

Destination

Directory: C:\Users\wnyoon\git\Test2

Initial branch: master

Clone submodules

Configuration

Remote name: origin

Projects

Import all existing Eclipse projects after clone finishes

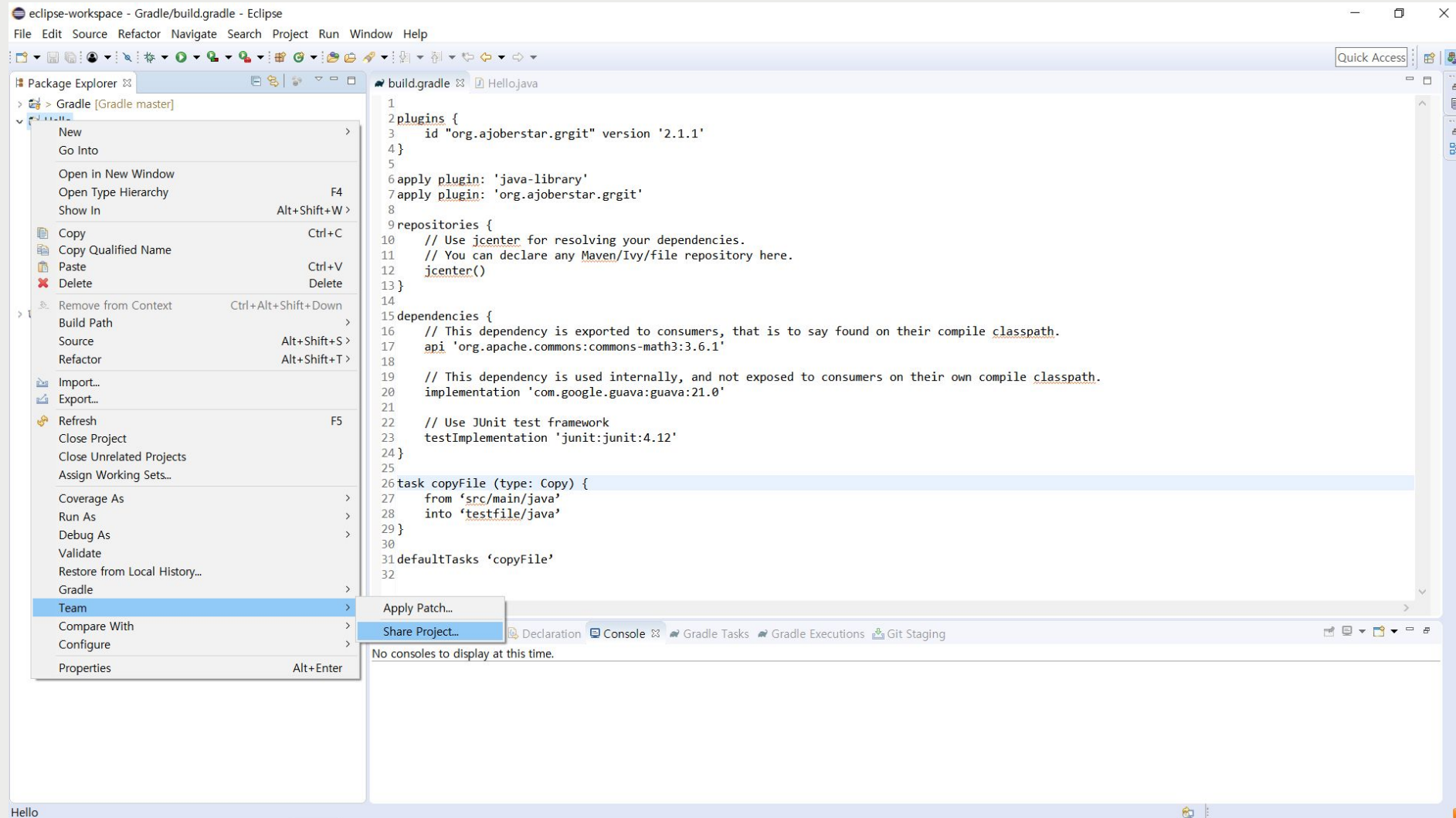
Working sets

Add project to working sets

Working sets:

| Commit

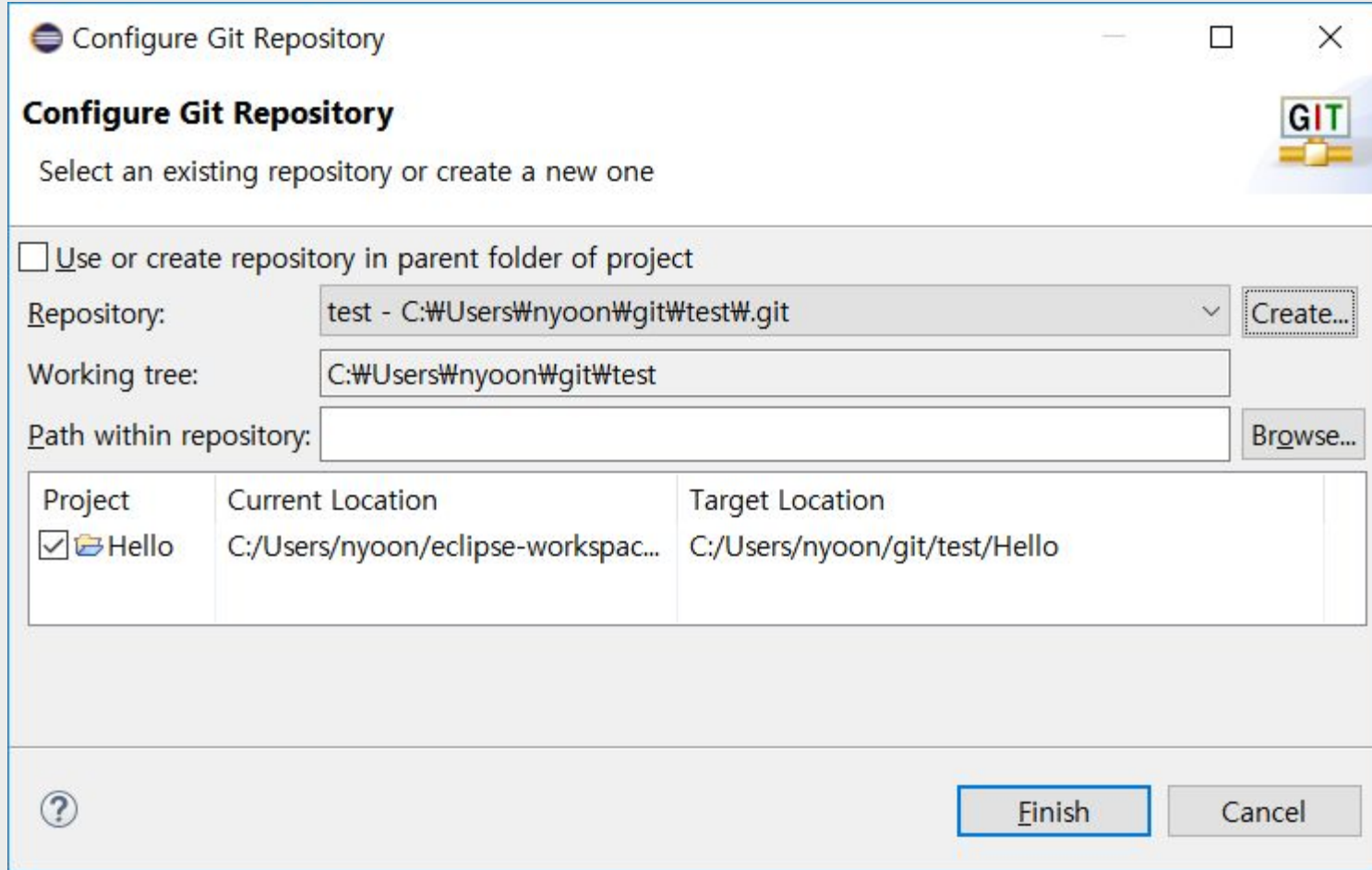
Package Explorer ▶ Team ▶ Share Project...



The screenshot shows the Eclipse IDE interface with the following components:

- Package Explorer:** Shows a project named 'Gradle' with a context menu open. The menu path is: Team > Share Project...
- Editor:** Displays the 'build.gradle' file with the following code:

```
1
2 plugins {
3   id "org.ajoberstar.grgit" version '2.1.1'
4 }
5
6 apply plugin: 'java-library'
7 apply plugin: 'org.ajoberstar.grgit'
8
9 repositories {
10  // Use jcenter for resolving your dependencies.
11  // You can declare any Maven/Ivy/file repository here.
12  jcenter()
13 }
14
15 dependencies {
16  // This dependency is exported to consumers, that is to say found on their compile classpath.
17  api 'org.apache.commons:commons-math3:3.6.1'
18
19  // This dependency is used internally, and not exposed to consumers on their own compile classpath.
20  implementation 'com.google.guava:guava:21.0'
21
22  // Use JUnit test framework
23  testImplementation 'junit:junit:4.12'
24 }
25
26 task copyFile (type: Copy) {
27   from 'src/main/java'
28   into 'testfile/java'
29 }
30
31 defaultTasks 'copyFile'
32
```
- Bottom Panel:** Shows 'No consoles to display at this time.' with tabs for Declaration, Console, Gradle Tasks, Gradle Executions, and Git Staging.

A dialog box titled "Configure Git Repository" with a Git logo in the top right corner. The main text says "Select an existing repository or create a new one". There is an unchecked checkbox for "Use or create repository in parent folder of project". Below this are three input fields: "Repository:" with a dropdown menu showing "test - C:\Users\nyoon\git\test\git" and a "Create..." button; "Working tree:" with a text box containing "C:\Users\nyoon\git\test"; and "Path within repository:" with an empty text box and a "Browse..." button. At the bottom is a table with three columns: "Project", "Current Location", and "Target Location". The table has one row with a checked checkbox, a folder icon, and the text "Hello". The "Current Location" is "C:/Users/nyoon/eclipse-workspac..." and the "Target Location" is "C:/Users/nyoon/git/test/Hello". At the very bottom are a help icon (?), a "Finish" button, and a "Cancel" button.

Configure Git Repository

Select an existing repository or create a new one

Use or create repository in parent folder of project

Repository: test - C:\Users\nyoon\git\test\git Create...

Working tree: C:\Users\nyoon\git\test

Path within repository: Browse...

Project	Current Location	Target Location
<input checked="" type="checkbox"/> Hello	C:/Users/nyoon/eclipse-workspac...	C:/Users/nyoon/git/test/Hello

? Finish Cancel



GitHub

Connect GitHub with Eclipse

GitHub

| Commit

Team ▶ Add to index

The screenshot shows the Eclipse IDE interface. The Package Explorer on the left shows a project structure with a 'Test' folder selected. A context menu is open over the 'Test' folder, with the 'Team' option selected. A sub-menu is open over 'Team', and the 'Add to Index' option is highlighted. The main editor shows a Java file named 'Hello.java' with the following code:

```
1 public class Hello {
2
3     public static void main(String args[]) {
4         System.out.println("Hello World!");
5     }
6 }
7
```

The bottom of the IDE shows the 'Git Staging' view with 'Changes (0)' and the 'Commit Message' dialog box. The 'Commit Message' dialog has the following fields:

Author: nyoon <nyoon@DESKTOP-TM6DB6A>
Committer: nyoon <nyoon@DESKTOP-TM6DB6A>

Buttons: Commit and Push..., Commit



GitHub

Connect GitHub with Eclipse

GitHub

| Commit

Team ▶ Commit...

The screenshot shows the Eclipse IDE interface. The Package Explorer on the left shows a project named 'Hello' with a 'test' folder. The main editor displays the 'Hello.java' file with the following code:

```
1 public class Hello {  
2     public static void main(String args[]) {  
3         System.out.println("Hello World!");  
4     }  
5 }
```

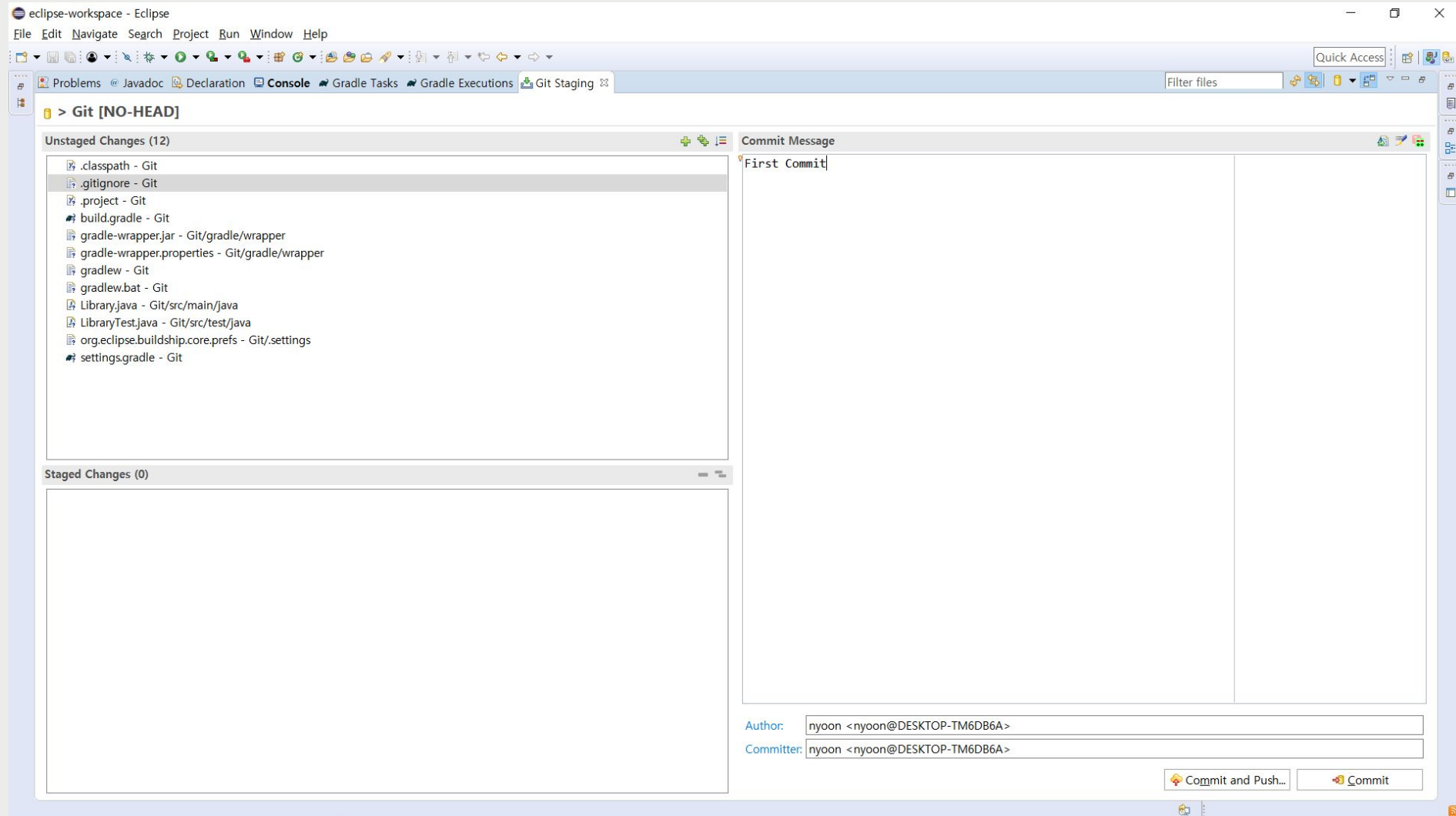
The 'Team' menu is open, and the 'Commit...' option is selected. The 'Commit Message' dialog is visible at the bottom, showing the following information:

- Staged Changes (17)
- Author: nyoon <nyoon@DESKTOP-TM6DB6A>
- Committer: nyoon <nyoon@DESKTOP-TM6DB6A>

Buttons for 'Commit and Push...' and 'Commit' are visible at the bottom right of the dialog.

| Commit

변경사항 확인 ▶ Commit Message 작성 ▶ Commit



Push Branch master

Push Confirmation

Confirm following expected push result.

master → master [new branch]

Message Details

Repository <https://github.com/SangHyeukYoon/Git>

Cancel push if result would be different than above because of changes on remote

Show dialog with result only when it is different from the confirmed result above

? < Back Next > Finish Cancel

| 확인

Git에서 Commit 결과를 확인할 수 있습니다.

The screenshot shows a web browser window displaying the GitHub repository page for 'SangHyeukYoon / Git'. The browser's address bar shows the URL 'https://github.com/SangHyeukYoon/Git/tree/master/Git'. The repository page includes navigation links for 'Code', 'Issues', 'Pull requests', 'Projects', 'Wiki', 'Insights', and 'Settings'. The current branch is 'master'. A commit history table is visible, showing a commit by 'nyoon and nyoon' with the message 'nyoon and nyoon 123' and a commit hash '491dc4a' made 'a minute ago'. The table lists various files and folders, all with a file size of '123' and a commit time of 'a minute ago'.

File/Folder	Size	Commit Time
nyoon and nyoon 123		Latest commit 491dc4a a minute ago
..		
.settings	123	a minute ago
gradle/wrapper	123	a minute ago
src	123	a minute ago
.classpath	123	a minute ago
.gitignore	123	a minute ago
.project	123	a minute ago
build.gradle	123	a minute ago
gradlew	123	a minute ago
gradlew.bat	123	a minute ago
settings.gradle	123	a minute ago



03

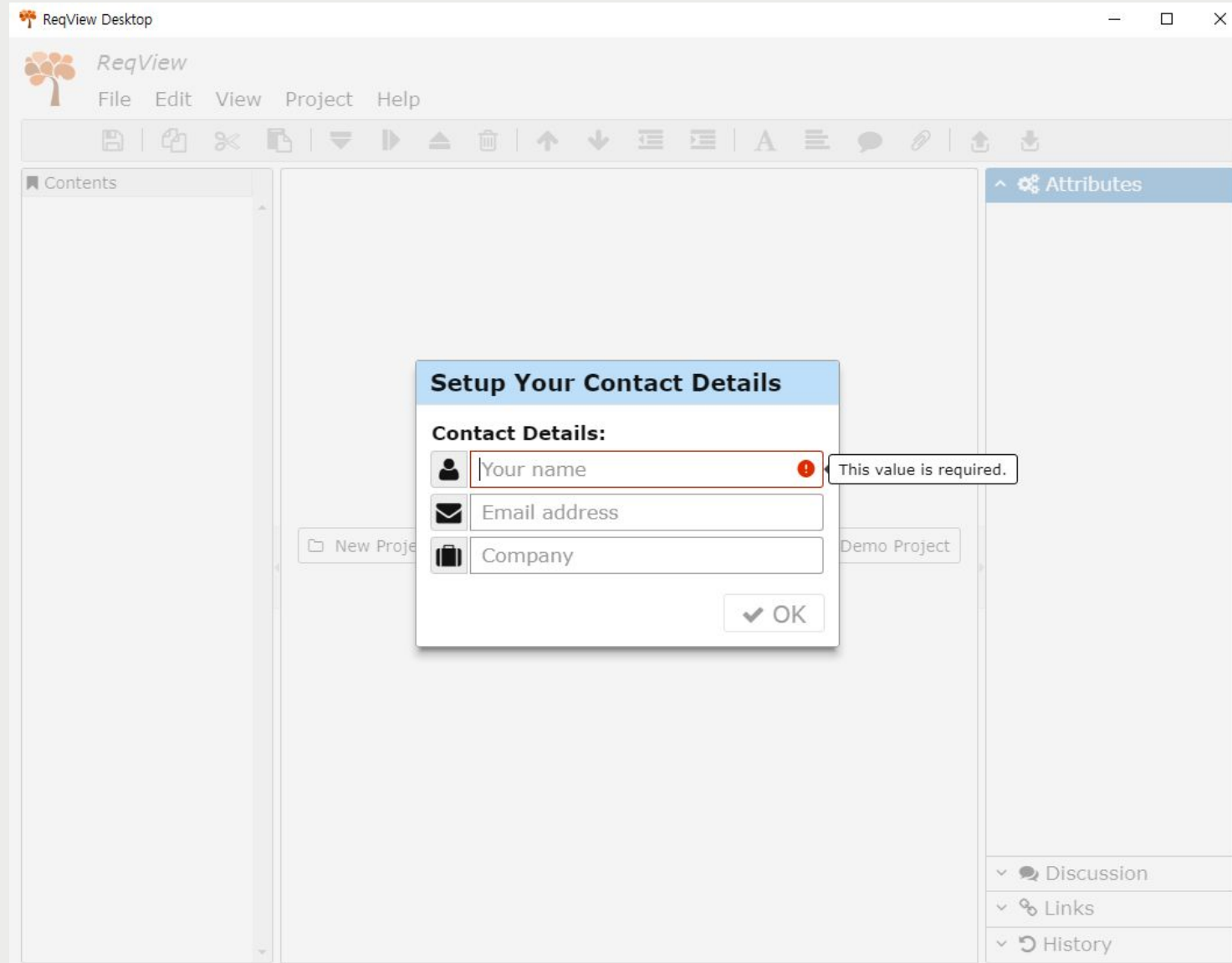
ReqView

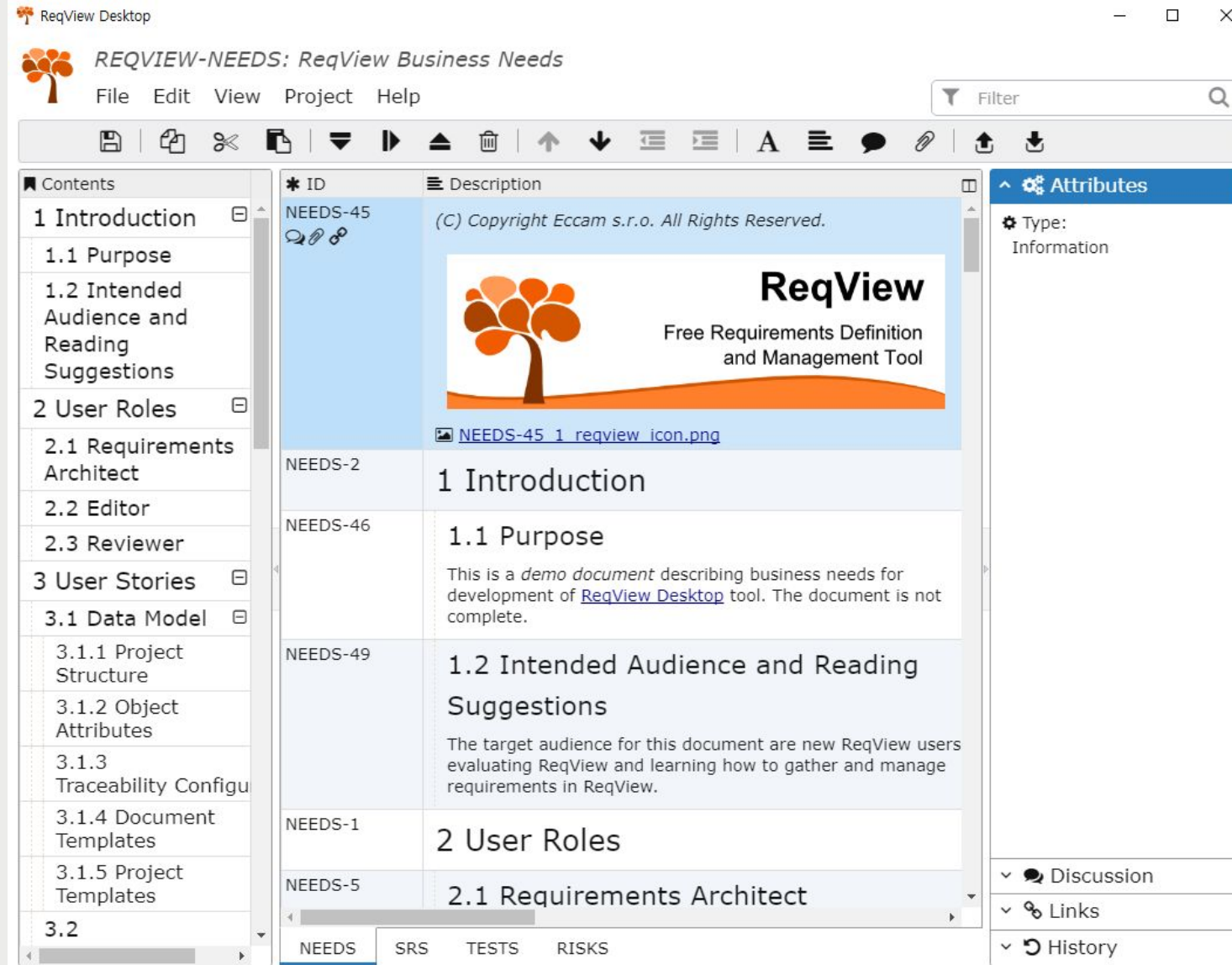
ReqView

Install



- ▶ 요구사항 관리 툴
- ▶ 트리구조 문서
- ▶ 각 문서마다 다양한 속성 정보 부여 가능 (예: 중요도, 저자 등)






The screenshot shows the ReqView Desktop application window. The title bar reads "ReqView Desktop". The main window title is "REQVIEW-NEEDS: ReqView Business Needs". The menu bar includes "File", "Edit", "View", "Project", and "Help". A search bar labeled "Filter" is in the top right. Below the menu is a toolbar with various icons for file operations and editing. The main content area is divided into three panes:

- Contents (Left):** A tree view showing the document structure:
 - 1 Introduction
 - 1.1 Purpose
 - 1.2 Intended Audience and Reading Suggestions
 - 2 User Roles
 - 2.1 Requirements Architect
 - 2.2 Editor
 - 2.3 Reviewer
 - 3 User Stories
 - 3.1 Data Model
 - 3.1.1 Project Structure
 - 3.1.2 Object Attributes
 - 3.1.3 Traceability Configur
 - 3.1.4 Document Templates
 - 3.1.5 Project Templates
 - 3.2

- Main Content (Center):** A table of requirements with columns for ID and Description.

* ID	Description
NEEDS-45	(C) Copyright Eccam s.r.o. All Rights Reserved. 
NEEDS-2	1 Introduction
NEEDS-46	1.1 Purpose This is a <i>demo document</i> describing business needs for development of ReqView Desktop tool. The document is not complete.
NEEDS-49	1.2 Intended Audience and Reading Suggestions The target audience for this document are new ReqView users evaluating ReqView and learning how to gather and manage requirements in ReqView.
NEEDS-1	2 User Roles
NEEDS-5	2.1 Requirements Architect
- Attributes (Right):** A panel showing "Attributes" with a gear icon and "Type: Information". At the bottom, there are expandable sections for "Discussion", "Links", and "History".

At the bottom of the application, there are tabs for "NEEDS", "SRS", "TESTS", and "RISKS".

The screenshot displays the ReqView Desktop application interface. The main window shows a table of requirements with columns for ID, Description, As a(n), I Want, and So that. The sidebar on the left contains a tree view of the project structure, with '3.1.2 Object Attributes' highlighted. A red box encloses the entire sidebar, and a red arrow points from the Korean text '요구사항 목록' to the highlighted item.

ID	Description	As a(n)	I Want	So that
NEEDS-19	3.1.2 Object Attributes	Architect	to be able to define custom object attributes	I can flexibly store additional information in project documents.
NEEDS-43	4.4 Number of Documents	Architect	to be able to define data model with many documents	I can store all in the project
NEEDS-13	3.2.1 Document Structure	Editor	to be able to organize document objects in a tree hierarchy with numbered sections	I can easily n structure
NEEDS-17	3.2.4 Image Attachments	Editor	to be able to attach images created in other tools	I can enhance with visual co diagrams, etc

요구사항 목록

- 1 Introduction
 - 1.1 Purpose
 - 1.2 Intended Audience and Reading Suggestions
- 2 User Roles
 - 2.1 Requirements Architect
 - 2.2 Editor
 - 2.3 Reviewer
- 3 User Stories
 - 3.1 Data Model
 - 3.1.1 Project Structure
 - 3.1.2 Object Attributes
 - 3.1.3 Traceability Configuration
 - 3.1.4 Document Templates
 - 3.1.5 Project Templates
 - 3.2 Requirements Definition
 - 3.2.1 Document Structure
 - 3.2.2 Rich Text Description
 - 3.2.3 Custom Attributes
 - 3.2.4 Image Attachments
 - 3.2.5 Document Attachments
 - 3.2.6 Document Object Links
 - 3.2.7 Project Files
 - 3.2.8 Autosave
 - 3.2.9 Filter
 - 3.2.10 Find
 - 3.2.11 Sort
 - 3.3 Review
 - 3.3.1 Traceability Browser
 - 3.3.2 Print
 - 3.3.3 Discussion

요구사항 내용

The screenshot displays the ReqView Desktop interface for a project named "REQVIEW-NEEDS: ReqView Business Needs". The main window is divided into several sections:

- Contents Panel (Left):** A tree view showing the project structure. The "3.1.2 Object Attributes" item is selected and highlighted.
- Table (Center):** A table with columns: ID, Description, As a(n), I Want, and So that. The table contains several rows of requirements, with "3.1.2 Object Attributes" (ID: NEEDS-19) highlighted in blue. Other rows include "4.4 Number of Documents" (ID: NEEDS-43) and "3.2.1 Document Structure" (ID: NEEDS-13).
- Attributes Panel (Right):** A panel titled "Attributes" showing details for the selected requirement. It lists attributes such as "As a(n): Architect", "I Want: to be able to define custom object attributes", and "So that: I can flexibly store additional information in project documents." It also lists "Acceptance Criteria" and "Tool supports" with a list of options like "number", "date", "boolean", "single and multivalued enumeration", and "string / rich formatted text".

A red box highlights the table and the attributes panel, with a red arrow pointing from the Korean text "요구사항 내용" to the table. A red line also connects the text to the top of the attributes panel.

ReqView Desktop

REQVIEW-NEEDS: ReqView Business Needs

File Edit View Project Help

Filter

Contents

- 1 Introduction
 - 1.1 Purpose
 - 1.2 Intended Audience and Reading Suggestions
- 2 User Roles
 - 2.1 Requirements Architect
 - 2.2 Editor
 - 2.3 Reviewer
- 3 User Stories
 - 3.1 Data Model
 - 3.1.1 Project Structure
 - 3.1.2 Object Attributes
 - 3.1.3 Traceability Configuration
 - 3.1.4 Document Templates
 - 3.1.5 Project Templates
 - 3.2 Requirements Definition
 - 3.2.1 Document Structure
 - 3.2.2 Rich Text Description
 - 3.2.3 Custom Attributes
 - 3.2.4 Image Attachments
 - 3.2.5 Document Attachments
 - 3.2.6 Document Object Links
 - 3.2.7 Project Files
 - 3.2.8 Autosave
 - 3.2.9 Filter
 - 3.2.10 Find
 - 3.2.11 Sort
 - 3.3 Review
 - 3.3.1 Traceability Browser
 - 3.3.2 Print
 - 3.3.3 Discussion

ID	Description	As a(n)	I Want	So that
NEEDS-19	3.1.2 Object Attributes	Architect	to be able to define custom object attributes	I can flexibly store additional information in project documents.
NEEDS-43	4.4 Number of Documents	Architect	to be able to define data model with many documents	I can store in the project
NEEDS-13	3.2.1 Document Structure	Editor	to be able to organize document objects in a tree hierarchy with numbered sections	I can easily n structure
NEEDS-17	3.2.4 Image Attachments	Editor	to be able to attach images created in other tools	I can enhance with visual co diagrams, etc

자세한 속성정보

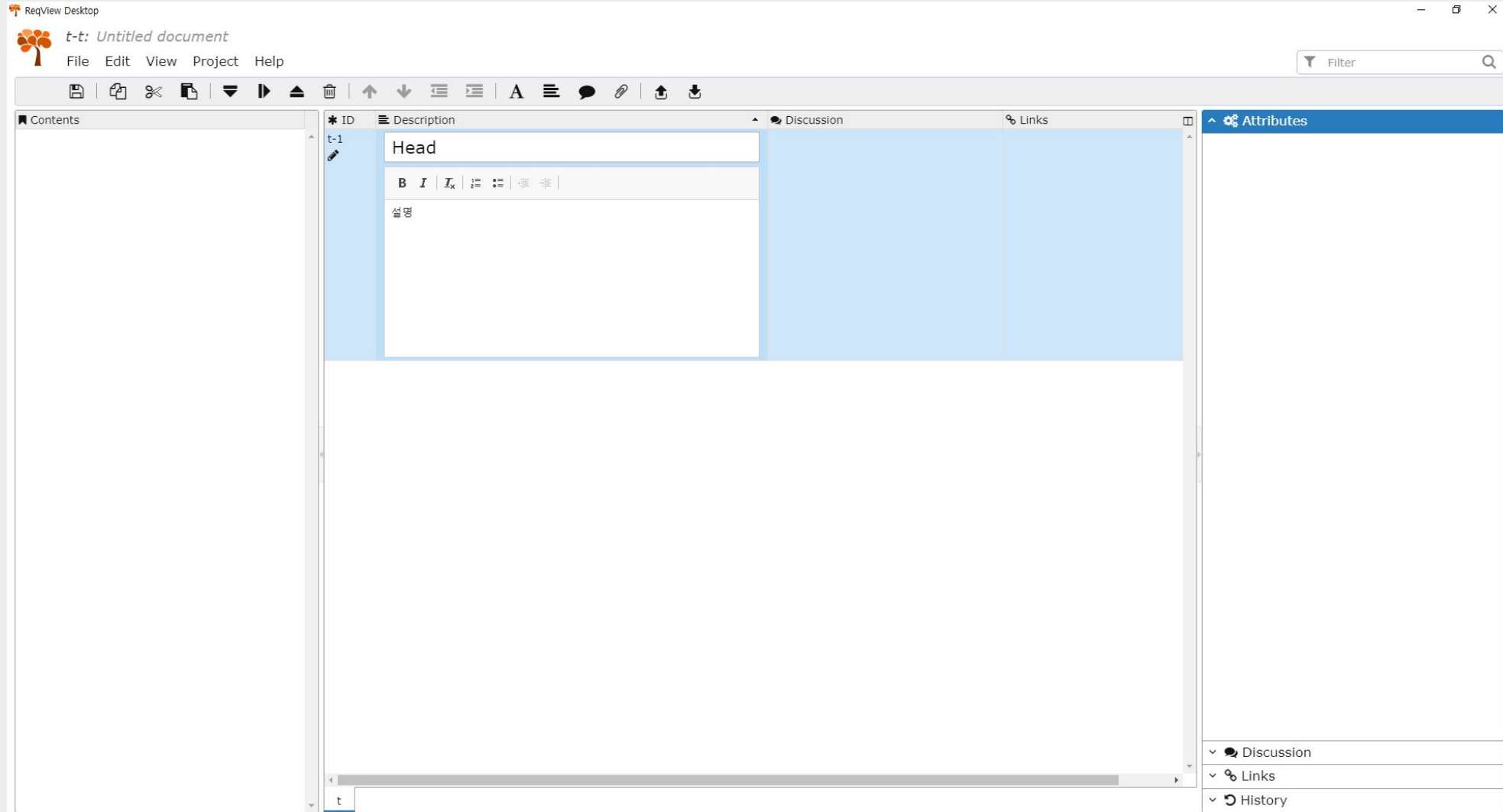
Attributes

- As a(n): Architect
- I Want: to be able to define custom object attributes
- So that: I can flexibly store additional information in project documents.
- Acceptance Criteria:
 - Tool supports:
 - creation of a new custom attribute,
 - deletion of a custom attribute,
 - definition of following custom attribute types
 - number
 - date
 - boolean
 - single and multivalued enumeration
 - string / rich formatted text
- Type: User Story

Discussion

Links

History



The screenshot displays the ReqView Desktop application window. The title bar reads "ReqView Desktop" and the document name is "t-t: Untitled document". The menu bar includes "File", "Edit", "View", "Project", and "Help". A search bar labeled "Filter" is located in the top right corner. The main workspace is divided into several sections:

- Contents:** A sidebar on the left showing a tree view with "1 Head".
- Table:** A central table with columns "ID", "Description", "Discussion", and "Links". It contains one row with ID "t-1" and Description "1 Head". The description field is currently selected, showing a small toolbar with icons for copy, paste, undo, redo, delete, and edit.
- Attributes:** A panel on the right side, currently empty.
- Bottom Panel:** A panel at the bottom right containing expandable sections for "Discussion", "Links", and "History".

The status bar at the bottom left shows the letter "t".

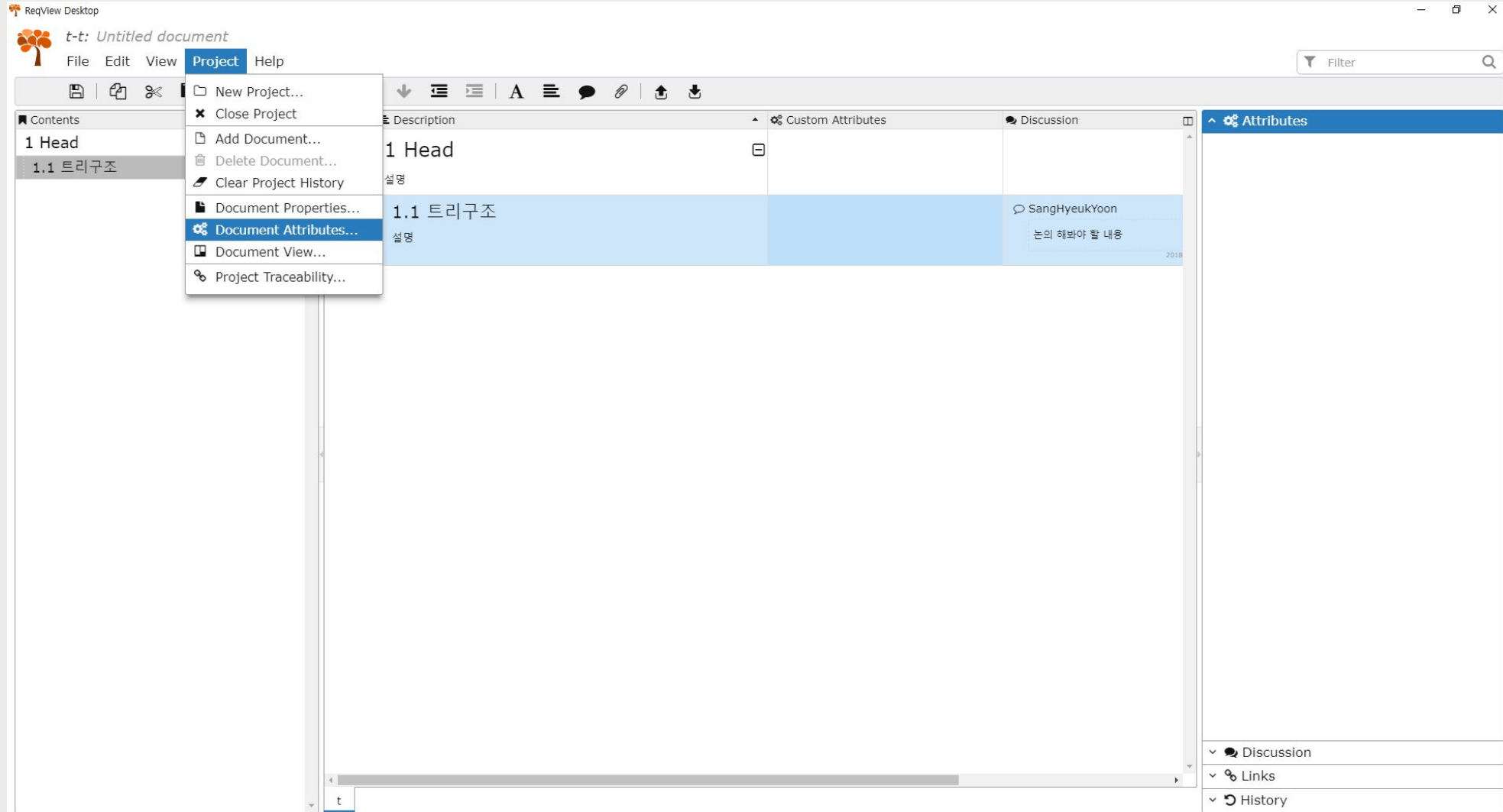
The screenshot displays the ReqView Desktop application window. The title bar reads "ReqView Desktop" and the document name is "t-t: Untitled document". The menu bar includes "File", "Edit", "View", "Project", and "Help". A toolbar with various icons is located below the menu bar. A search filter box is positioned in the top right corner.

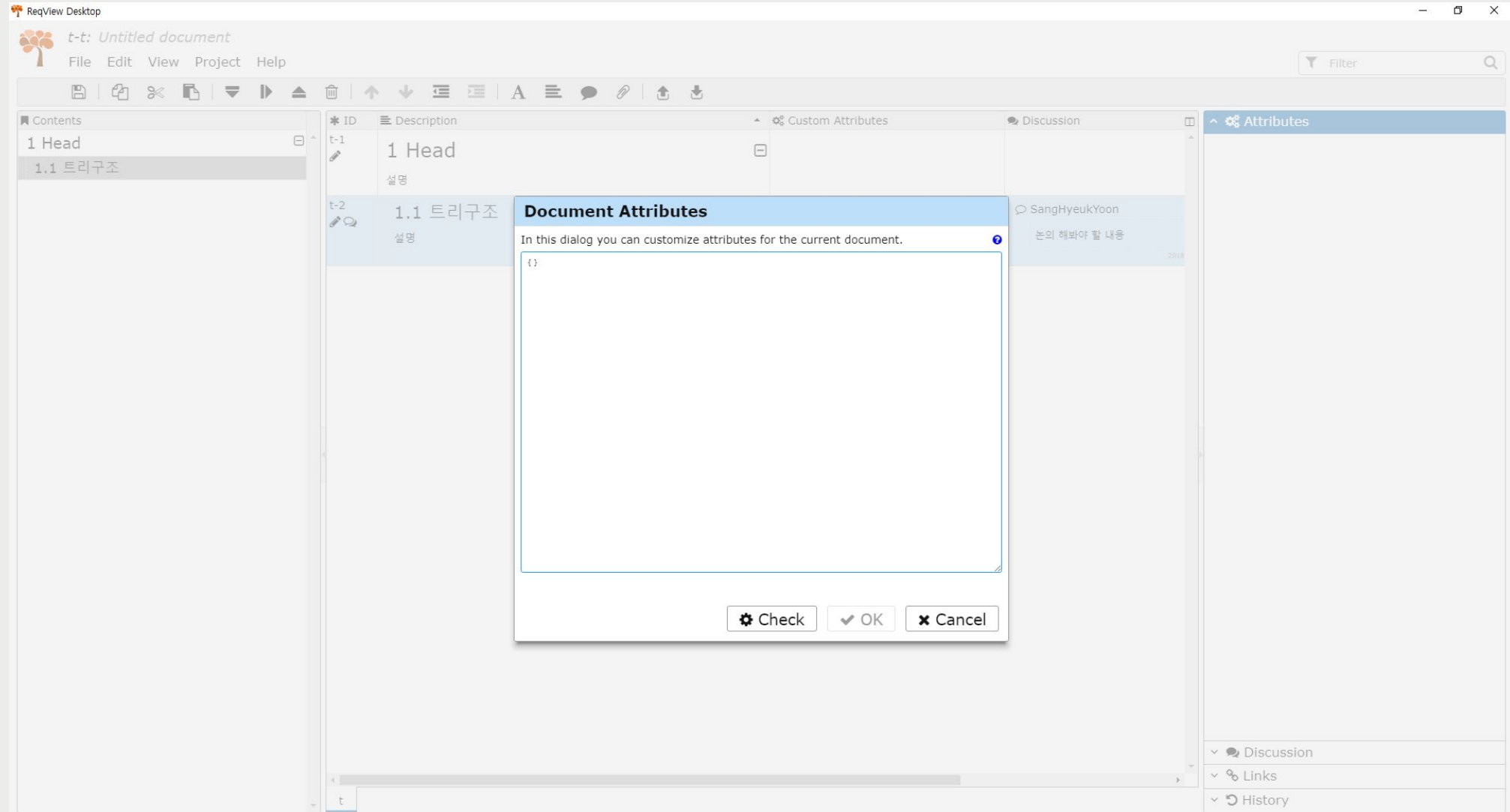
The main workspace is divided into several panels:

- Contents:** A sidebar on the left showing a tree structure with a single item "1 Head".
- Description:** A central panel showing the details of the selected item. It contains a table with the following content:

ID	Description	Discussion	Links
t-1	1 Head		
t-2	트리구조		
- Attributes:** A panel on the right side, currently empty.

At the bottom right, there is a vertical stack of expandable sections: "Discussion", "Links", and "History".





| 속성 추가

ReqView Desktop

t-t: Untitled document

File Edit View Project Help

Filter

Contents

- 1 Head
- 1.1 트리구조

* ID Description

- t-1 1 Head
- t-2 1.1 트리구조

Custom Attributes Discussion

Attributes

Document Attributes

In this dialog you can customize attributes for the current document.

```
{
  "status": {
    "name": "Status",
    "type": "enum",
    "values": [
      {
        "key": "Unknown"
      },
      {
        "key": "Draft"
      },
      {
        "key": "Accepted"
      },
      {
        "key": "Rejected"
      },
      {
        "key": "Implemented"
      }
    ]
  }
}
```

Check OK Cancel

Discussion

Links

History

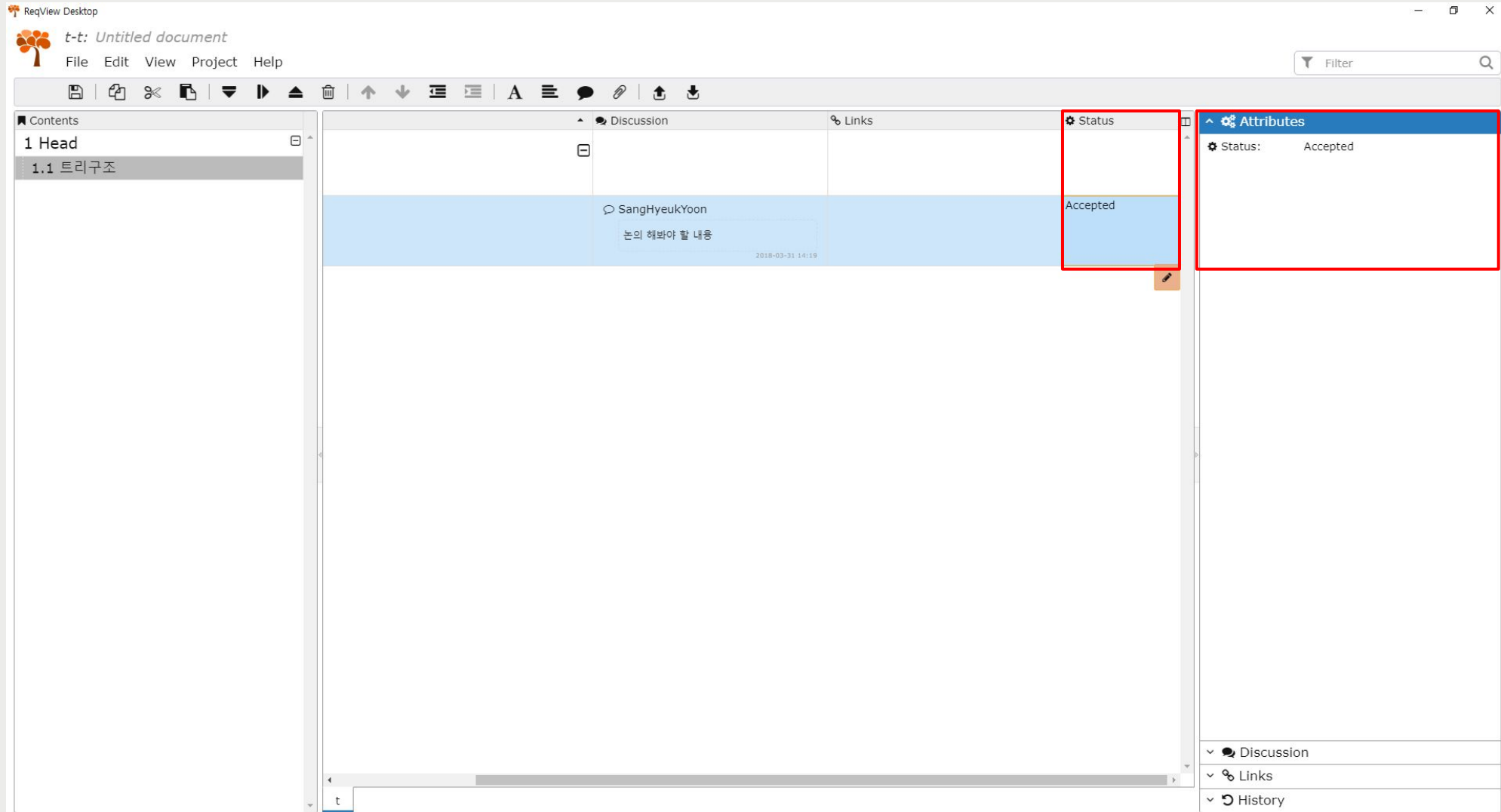
```
{  
  "status": {  
    "name": "Status",  
    "type": "enum",  
    "values": [  
      { "key": "Unknown" },  
      { "key": "Draft" },  
      { "key": "Accepted" },  
      { "key": "Rejected" },  
      { "key": "Implemented" },  
      { "key": "Verified" } ]  
    }  
  }  
}
```

The screenshot displays the ReqView Desktop application window. The title bar reads "ReqView Desktop" and the document is titled "t-t: Untitled document". The menu bar includes "File", "Edit", "View", "Project", and "Help". A toolbar with various icons is located below the menu. The main workspace is divided into several panels:

- Contents:** A tree view on the left showing a hierarchy with "1 Head" and "1.1 트리구조".
- Discussion:** A central panel showing a comment by "SangHyeukYoon" with the text "논의 해봐야 할 내용" and a timestamp of "2018-03-31 14:19".
- Links:** A panel for managing links.
- Status:** A panel for document status.
- Attributes:** A panel on the right for adding and editing attributes.

At the bottom right, there is a vertical sidebar with expandable sections for "Discussion", "Links", and "History".

The screenshot displays the ReqView Desktop application window. The title bar reads "ReqView Desktop" and the document is titled "t-t: Untitled document". The menu bar includes "File", "Edit", "View", "Project", and "Help". A toolbar with various icons is located below the menu bar. On the left, a "Contents" pane shows a tree structure with "1 Head" and "1.1 트리구조". The main workspace is divided into several panes: "Discussion", "Links", "Status", and "Attributes". A discussion comment by "SangHyeukYoon" with the text "논의 해봐야 할 내용" and a timestamp of "2018-03-31 14:19" is highlighted. A dropdown menu is open over the "Status" column, showing options: "none", "Unknown", "Draft", "Accepted", "Rejected", "Implemented", and "Verified". The "Attributes" pane is currently empty. At the bottom right, a sidebar contains expandable sections for "Discussion", "Links", and "History".



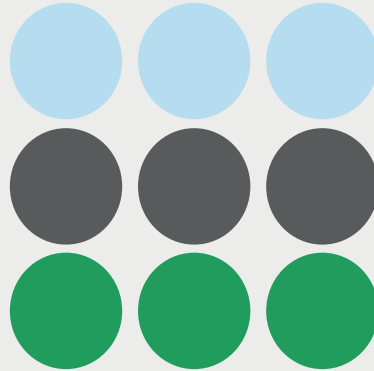


04

CTIP

CTIP





THANK YOU

SOFTWARE VERIFICATION