

영상관리

2019 – (주)텔코코리아아이에스

2022 – (주)리모샷

2023 – 넥스트IT

강사: 서지민

연락처: 010-3376-1886

메일: jmseokm29a1@naver.com

<https://github.com/RadeonHD6750>



일상생활

원격 저장소



넥스트 IT 교육센터

여기서 작업하던 것을

학원에서도 그대로 이어서 가능



집에서도 그대로 이어서 가능

집에서 주말에 했던 것을

일상생활

원격 저장소



시간과 공간에 제약이 없다.

넥스트 IT 교육센터

여기서 작업하던 것을

학원에서도 그대로 이어서 가능

네이버, 구글 클라우드와 무슨 차이??



집에서도 그대로 이어서 가능

집에서 주말에 했던 것을

프로젝트 관리 개념

◆ 정의

프로젝트의 진행상황 및 변경사항 등을 체계적으로 추적하고 통제하는 것

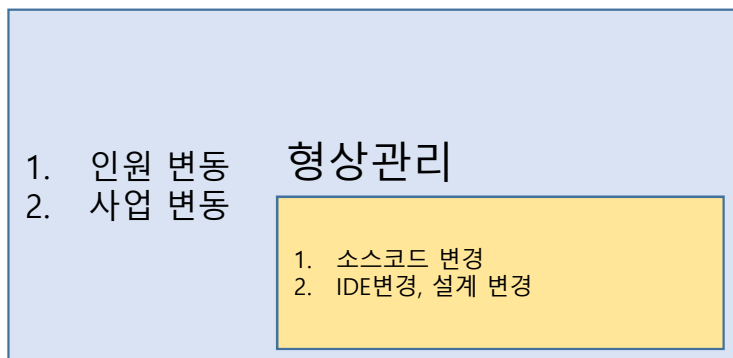
◆ 프로젝트의 변경사항 종류

- 소스코드 변경사항 -> 깃으로 관리
- IDE, 빌드구조, 설계 변동 등등
- 참여인원에 대한 변동 (근태, 인원수 변동, 인력 보충 등)
- 사업에 대한 변동 (요구사항 변동, 일정 변동 등)

◆ 용어 정리

- 변경관리 : 소스코드 변경사항에 대한 관리
- 버전관리 : 변경사항을 '버전'이란 개념을 통하여 관리
- 형상관리 : 위의 개념들을 포함하여 프로젝트와 관련된 모든 변경사항을 관리

프로젝트 관리



프로젝트 형상관리 필요성

◆ 필요성 - 협업

프로젝트는 혼자서도 하나 협업에서는 최소 2명 이상의 팀으로 움직인다. 누가 어느 부분을 얼마나 작업 및 수정하였는지 파악하기가 쉽지 않다. 이는 혼자 진행하는 경우도 비슷하게 본인이 언제 어느 부분을 얼마나 작업 및 수정하였는지 일일이 기억하기 쉽지 않다.

따라서 별도의 절차와 관리도구를 가지고 프로젝트의 형상을 관리한다.

◆ 적용범위 - 소스코드

주로 소프트웨어 분야에서 프로젝트 형상관리 하는 것처럼 보일 수 있으나 거진 모든 분야의 프로젝트에서 형상관리를 하고 있다.

허나 여기서는 소프트웨어를 다루고 있기에 그 범위를 줄여서 ‘소스코드’ 관리에 초점을 두는 바이다.

형상관리 도구 비교

	SVN	Git	추가설명
저장소 방식	중앙집중형	분산형	
히스토리	중앙식	개인보관	
Commit	즉시반영	사후반영	각 개인 저장소 비교
네트워크	완전 온라인	부분 온라인	
중앙서버 복구	로컬 복구 불가	로컬 복구 가능	구성원 중 하나라도 있으면 됨
장점	직관적 운용	로컬 관리, 고속동작	
단점	부하취약, 소실위험	공부가 필요	
점유율	사향	압도적	https://rhodecode.com/insights/version-control-systems-2016
업계동향 사용처	일부 게임, 하드웨어	소프트웨어 SI, 솔루션	
적합한 특성	뭔가 무거운 것	자주 변경되는 것	

Git Client 설치하기



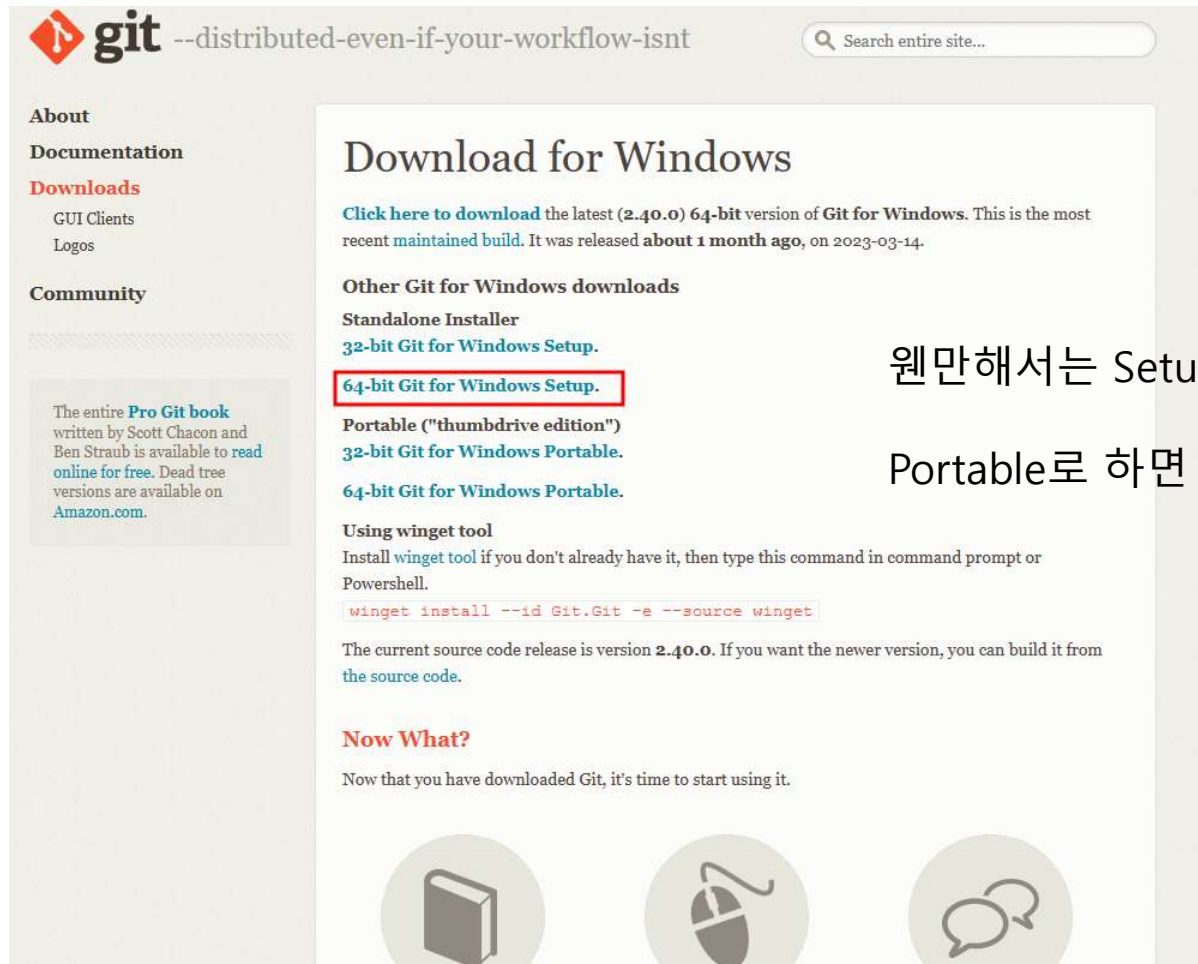
Git 설치 - 리눅스

```
jimin@jimin-meat: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)

jimin@jimin-meat:~$ git
명령어 'git' 을(를) 찾을 수 없습니다. 그러나 다음을 통해 설치할 수 있습니다:
sudo apt install git
jimin@jimin-meat:~$ sudo apt install git
[sudo] jimin 암호:
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다... 완료
상태 정보를 읽는 중입니다... 완료
다음의 추가 패키지가 설치될 것입니다 :
  git-man liberror-perl
제안하는 패키지:
  git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb
  git-cvs git-mediawiki git-svn
다음 새 패키지를 설치할 것입니다:
  git git-man liberror-perl
0개 업그레이드, 3개 새로 설치, 0개 제거 및 224개 업그레이드 안 함.
4,121 k바이트 아카이브를 받아야 합니다.
이 작업 후 20.9 M바이트의 디스크 공간을 더 사용하게 됩니다.
계속 하시겠습니까? [Y/n] Y
받기:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 liberror-perl all 0.170
29-1 [26.5 kB]
받기:2 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 git-man all 1:2
.34.1-1ubuntu1.8 [953 kB]
```

1. Git이 있는지 터미널에 입력하여 존재여부 확인
2. 없다면 `sudo apt install git`으로 설치

Git 설치 - 윈도우



웹만해서는 Setup버전을 하자

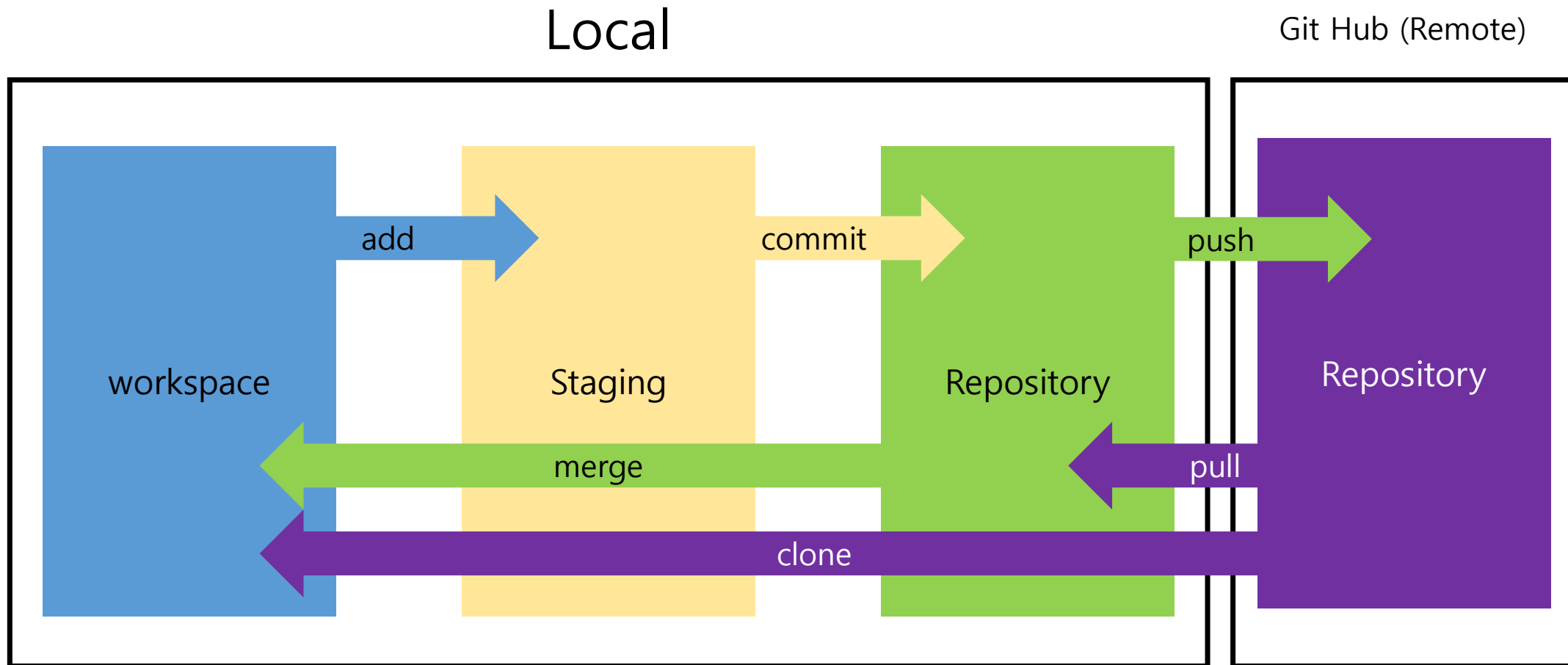
Portable로 하면 나중에 못 찾을

<https://git-scm.com/download/win>

GitHub 저장소 생성



Git 구조도



GitHub 저장소 생성

The screenshot shows the GitHub homepage. At the top, there's a navigation bar with the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. On the left sidebar, under 'Recent Repositories', there's a 'New' button highlighted with a red box. Next to it, the text 'New 클릭' is written in red. Below this, a list of repositories is shown, including CUJeong/Memory, CUJeong/BigData, CUJeong/EndGame, CUJeong/GameFactory, CUJeong/JavaJeong, and CUJeong/NextItEveryTime. The main content area has tabs for 'Following' and 'For you' (marked as Beta). A welcome message is displayed, followed by an 'Explore GitHub' section. On the right, there are two promotional cards: one for GitHub Copilot and another about GitHub's response to the war in Ukraine. At the bottom, there's a footer with links for Blog, API, Terms, About, Training, Privacy, Shop, Status, Docs, Contact GitHub, Security, and Pricing.

GitHub 저장소 생성



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *

CUJeong ▼

Repository name *

MyProject ✓

프로젝트명 설정

Great repository names are short and memorable. Need inspiration? How about [didactic-winner](#)?

Description (optional)

☒ Public

Anyone on the internet can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file



This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▼

GitHub 저장소 생성

- ☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.
- ☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

- ☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

**프로젝트 설명 파일을
따로 추가하려면 체크**

Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▾

**Commit시 자동으로 제외될
항목들을 추가하려면 체크**

Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

License: None ▾

 You are creating a public repository in your personal account.

Create repository

Create repository 클릭

GitHub 저장소 생성

The screenshot shows the GitHub interface for a repository named 'RadeonHD6750 / JavaExample'. The repository is public and has 1 branch and 0 tags. The 'Code' dropdown menu is open, showing options to clone the repository using HTTPS, SSH, or GitHub CLI. The HTTPS URL is highlighted in a red box: `https://github.com/RadeonHD6750/JavaExample.g`. Other options include 'Open with GitHub Desktop' and 'Download ZIP'. The repository's file list is visible on the left, showing folders like 'Libraries', 'ch01_helloworld', 'ch02_variable', 'ch03_operator', 'ch04_standardIO', 'ch05_controll', and 'ch06_method'. The right sidebar shows the 'About' section with details about the repository's README, license (GPL-3.0), and stars (0).

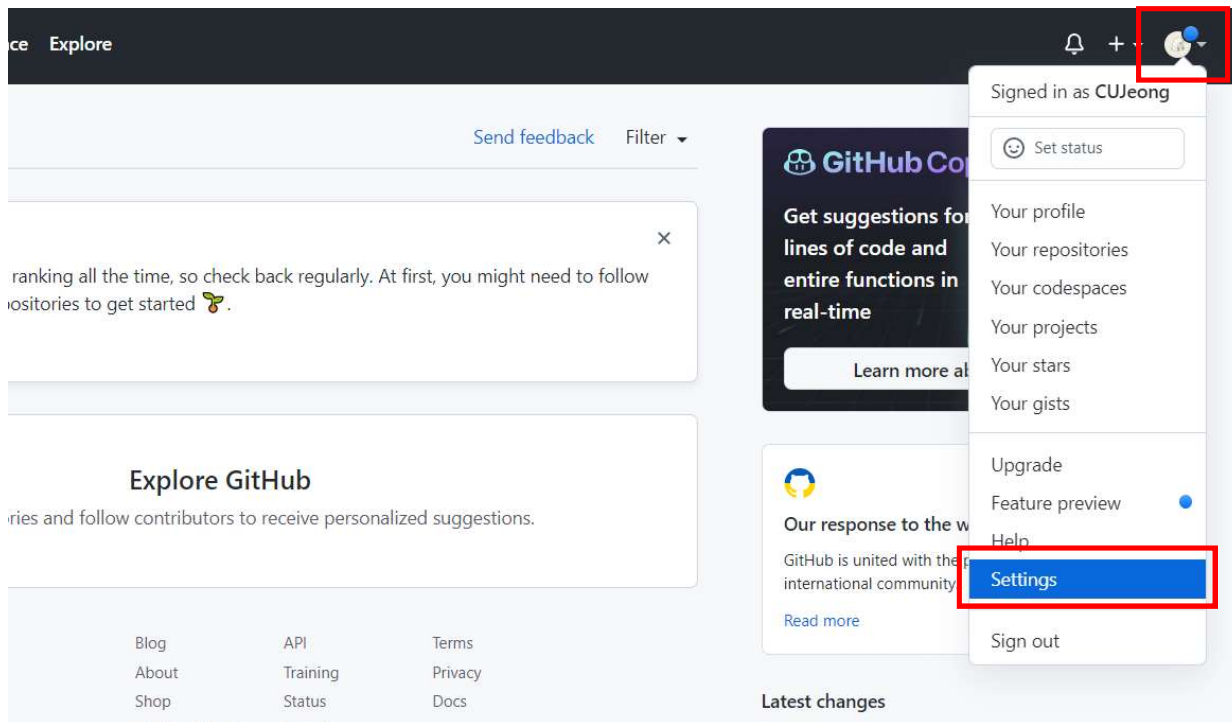
Git 주소 취득

GitHub 토큰 생성



GitHub 토큰 생성

◆ GitHub 토큰은 VS Code나 Eclipse에서 GitHub에 로그인 시 사용되는 비밀번호라고 보면 된다.



프로필 이미지 클릭

Settings 클릭

GitHub 토큰 생성



Go to your personal profile

- Public profile
- Account
- Appearance
- Accessibility
- Notifications
- Access
- Billing and plans
- Emails
- Password and authentication
- SSH and GPG keys
- Organizations
- Moderation

스크롤 내리기



- Code, planning, and automation
- Repositories
- Packages
- GitHub Copilot

Public profile

Name

Your name may appear around GitHub where you contribute or are mentioned. You can remove it at any time.

Public email

You have set your email address to private. To toggle email privacy, go to [email settings](#) and uncheck "Keep my email address private."

Bio

You can @mention other users and organizations to link to them.

URL

Twitter username

Profile picture



Edit

GitHub 토큰 생성

Security

🔒 Code security and analysis

Integrations

🔌 Applications

🕒 Scheduled reminders

Archives

📖 Security log

📖 Sponsorship log

⏏ Developer settings

Developer settings
클릭

You can @mention your company's GitHub organization to link it.

Location

All of the fields on this page are optional and can be deleted at any time, and by filling them out, you're giving us consent to share this data wherever your user profile appears. Please see our [privacy statement](#) to learn more about how we use this information.

Update profile

Contributions & Activity

☐ Make profile private and hide activity Beta

Enabling this will hide your contributions and activity from your GitHub profile and from social features like followers, stars, feeds, leaderboards and releases.

☐ Include private contributions on my profile

Your contribution graph, achievements, and activity overview will show your private contributions without revealing any repository or organization information. [Read more](#).


Update preferences

Profile settings

☒ Show Achievements on my profile

Your achievements will be shown on your profile.

GitHub 토큰 생성



[Pull requests](#)
[Issues](#)
[Marketplace](#)
[Explore](#)

[Settings](#) / Developer settings

 GitHub Apps

 OAuth Apps

 **Personal access tokens**

GitHub Apps

New GitHub App

Want to build something that integrates with and extends GitHub? [Register a new GitHub App](#) to get started developing on the GitHub API. You can also read more about building GitHub Apps in our [developer documentation](#).

Personal access tokens 클릭



© 2022 GitHub, Inc.

[Terms](#)

[Privacy](#)

[Security](#)

[Status](#)

[Docs](#)

[Contact GitHub](#)

[Pricing](#)

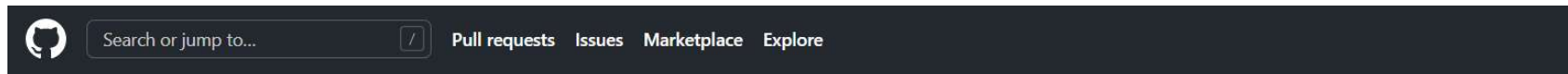
[API](#)

[Training](#)

[Blog](#)

[About](#)

GitHub 토큰 생성



[Settings](#) / [Developer settings](#)

GitHub Apps

OAuth Apps

Personal access tokens

Personal access tokens

Generate new token

Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

Generate new token 클릭

GitHub 토큰 생성



[Settings](#) / [Developer settings](#)

- GitHub Apps
- OAuth Apps
- Personal access tokens

New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

myproject_token

토큰 이름 설정

What's this token for?

Expiration *

90 days

The token will expire on Mon, Oct 31 2022

토큰 유효기간 설정

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows

GitHub 토큰 생성

Select scopes 는 이 토큰을 통해 로그인했을시 이용 가능한 권한을 설정한다고 보면 된다.

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input checked="" type="checkbox"/> workflow	Update GitHub Action workflows
<input checked="" type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input checked="" type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input checked="" type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry
<input checked="" type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects
<input checked="" type="checkbox"/> write:org	Read and write org and team membership, read and write org

스크롤 내리기

↓

<input checked="" type="checkbox"/> admin:gpg_key	Full control of public user GPG keys (Developer Preview)
<input checked="" type="checkbox"/> write:gpg_key	Write public user GPG keys
<input checked="" type="checkbox"/> read:gpg_key	Read public user GPG keys

Generate token

Cancel

Generate token 클릭

GitHub 토큰 생성

Personal access tokens

Generate new token

Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your new personal access token now. You won't be able to see it again!



토큰 생성됨



Delete

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API](#) over [Basic Authentication](#).

**생성된 토큰을 VS Code나 Eclipse에서 로그인할 때 언제나 사용할 수 있도록
따로 복사해서 저장해놓는다. (로컬파일 뿐 아니라 본인 메일에도 저장하는 걸 추천)**




생성했던 토큰을 잊어버렸다면 그냥 다시 발급받으면 된다.

기존 프로젝트에 git 붙이기



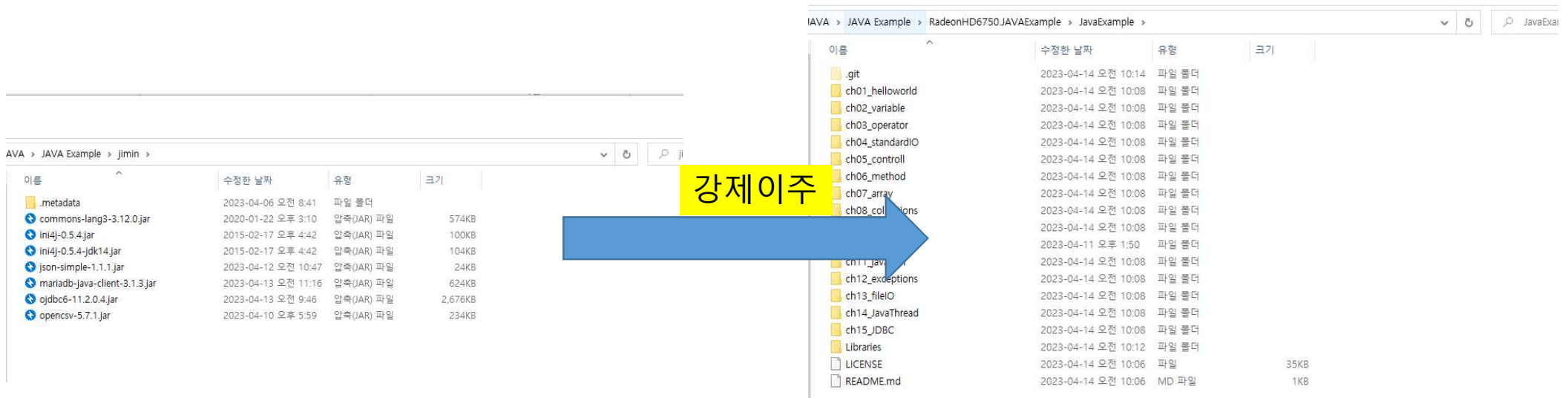
프로젝트 백업

JAVA > JAVA Example

이름	수정된 날짜	유형	크기
JavaExample	2023-04-14 오후 4:38	파일 폴더	
jeong	2023-04-04 오후 2:05	파일 폴더	
jimin	2023-04-14 오전 10:08	파일 폴더	
RadeonHD6750.JAVAExample	2023-04-14 오전 10:06	파일 폴더	
 JAVA Example-2023-04-14.zip	2023-04-14 오전 9:25	압축(ZIP) 파일	2,424,598...
 JAVA Example-2023-04-17.zip	2023-04-17 오후 5:38	압축(ZIP) 파일	126,072KB
 jimin.zip	2023-04-14 오전 9:46	압축(ZIP) 파일	20,681KB

본 과정을 진행하기에 앞서 반드시 프로젝트 폴더를 통째로 압축하여
앞으로의 위험에 대비하도록 하자.

주의사항



강제이주

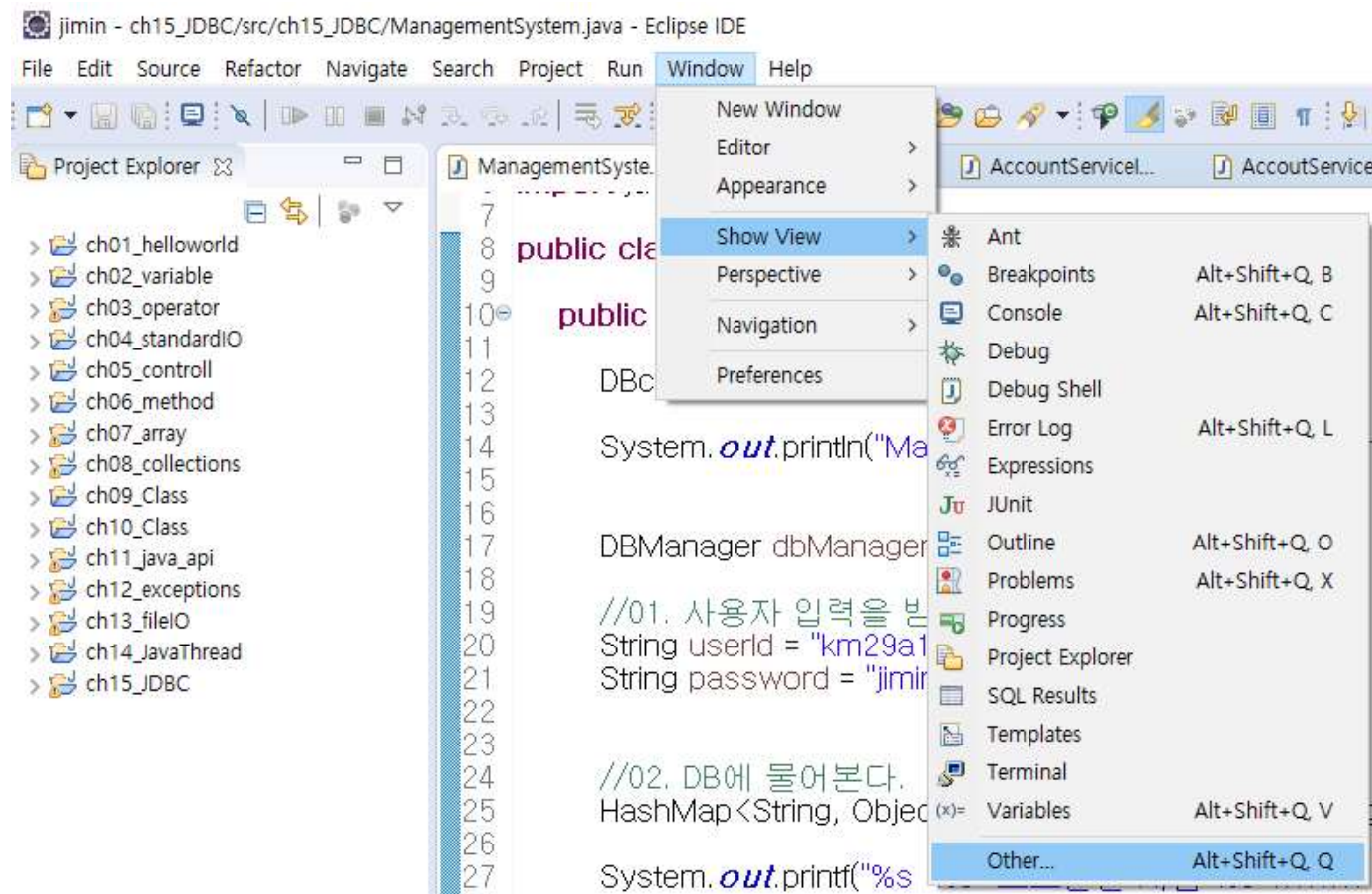
이름	수정한 날짜	유형	크기
.metadata	2023-04-06 오전 8:41	파일 폴더	
commons-lang3-3.12.0.jar	2020-01-22 오후 3:10	압축(JAR) 파일	574KB
ini4j-0.5.4.jar	2015-02-17 오후 4:42	압축(JAR) 파일	100KB
ini4j-0.5.4-jdk14.jar	2015-02-17 오후 4:42	압축(JAR) 파일	104KB
json-simple-1.1.1.jar	2023-04-12 오전 10:47	압축(JAR) 파일	24KB
mysql-connector-java-8.0.29.jar	2023-04-13 오전 11:16	압축(JAR) 파일	624KB
ojdbc6-11.2.0.4.jar	2023-04-13 오전 9:46	압축(JAR) 파일	2,676KB
opencsv-5.7.1.jar	2023-04-10 오후 5:59	압축(JAR) 파일	234KB

이름	수정한 날짜	유형	크기
.git	2023-04-14 오전 10:14	파일 폴더	
ch01_helloworld	2023-04-14 오전 10:08	파일 폴더	
ch02_variable	2023-04-14 오전 10:08	파일 폴더	
ch03_operator	2023-04-14 오전 10:08	파일 폴더	
ch04_standardIO	2023-04-14 오전 10:08	파일 폴더	
ch05_controll	2023-04-14 오전 10:08	파일 폴더	
ch06_method	2023-04-14 오전 10:08	파일 폴더	
ch07_array	2023-04-14 오전 10:08	파일 폴더	
ch08_collections	2023-04-14 오전 10:08	파일 폴더	
ch11_java	2023-04-14 오전 10:08	파일 폴더	
ch12_exceptions	2023-04-14 오전 10:08	파일 폴더	
ch13_fileIO	2023-04-14 오전 10:08	파일 폴더	
ch14_JavaThread	2023-04-14 오전 10:08	파일 폴더	
ch15_JDBC	2023-04-14 오전 10:08	파일 폴더	
Libraries	2023-04-14 오전 10:12	파일 폴더	
LICENSE	2023-04-14 오전 10:06	파일	35KB
README.md	2023-04-14 오전 10:06	MD 파일	1KB

원래 있던 프로젝트 폴더가 통째로 새로 지정한 저장소로 강제 이주 됨.

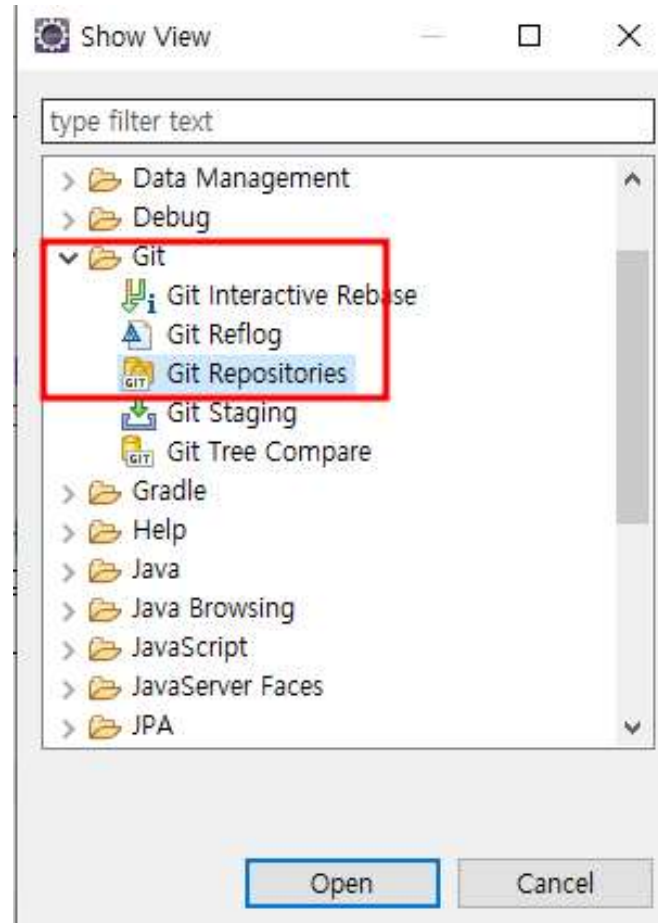
고로 처음부터 로컬 저장소를 만들고 거기서 프로젝트를 만들어야 한다.

이클립스에 git 붙이기



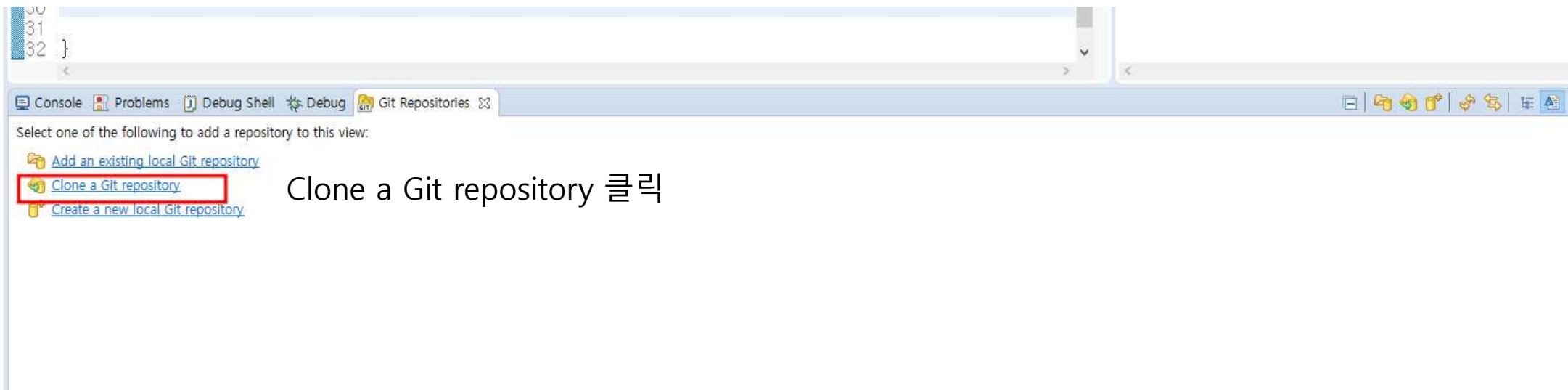
Window -> show view -> other 클릭

이클립스에 git 붙이기



Git -> Git Repositories -> Open 클릭

이클립스에 git 붙이기



이클립스에 git 붙이기



The image shows the 'Clone Git Repository' dialog box in Eclipse. It has a title bar with a gear icon and the text 'Clone Git Repository'. Below the title bar is a section titled 'Source Git Repository' with the instruction 'Enter the location of the source repository.' and a small Git logo. The dialog is divided into three main sections: 'Location', 'Connection', and 'Authentication'. In the 'Location' section, there are fields for 'URI' (containing 'https://github.com/RadeonHD6750/JavaExample.git'), 'Host' (containing 'github.com'), and 'Repository path' (containing '/RadeonHD6750/JavaExample.git'). There is a 'Local File...' button next to the URI field. In the 'Connection' section, there is a 'Protocol' dropdown menu set to 'https' and an empty 'Port' field. In the 'Authentication' section, there are fields for 'User' and 'Password'. The 'User' field is highlighted with a yellow box and contains the text '이메일 주소'. The 'Password' field is highlighted with a yellow box and contains the text '발행한 토큰' followed by five black dots. There is a checkbox labeled 'Store in Secure Store' which is unchecked. At the bottom of the dialog, there are four buttons: a help button (question mark icon), '< Back', 'Next >' (which is highlighted with a blue border), 'Finish', and 'Cancel'.

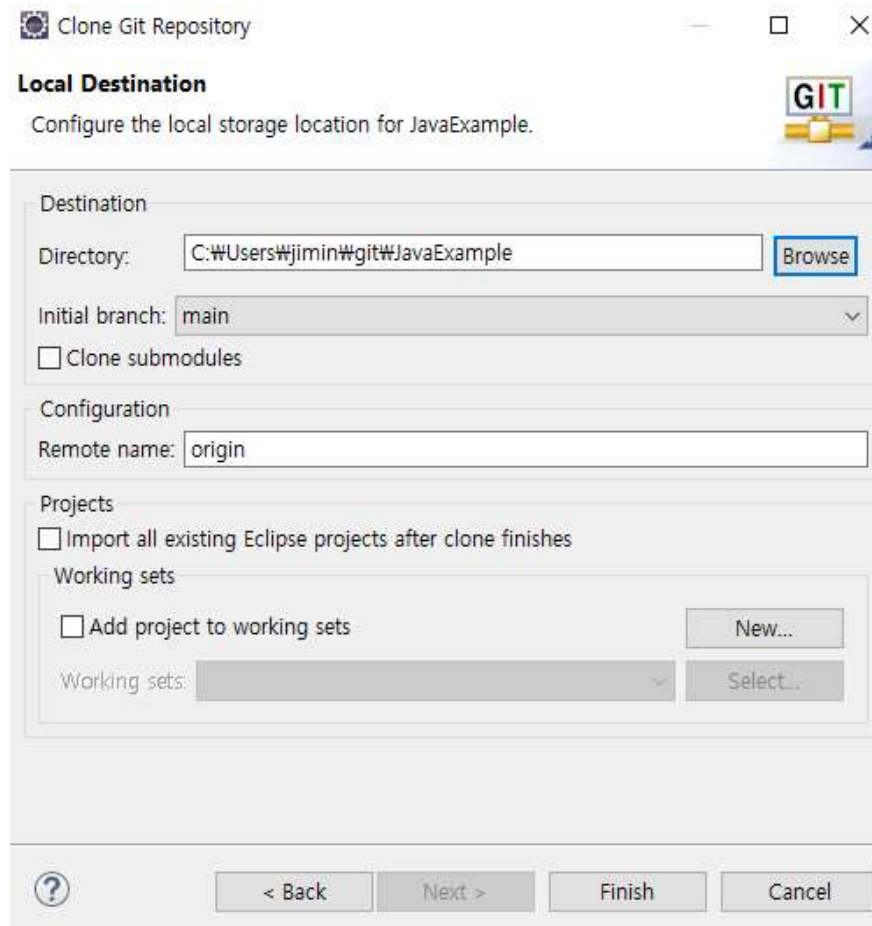
아까 취득한 git 주소 복붙

하면 나머지 알아서 자동완성됩니다.

이메일 주소

발행한 토큰

이클립스에 git 붙이기

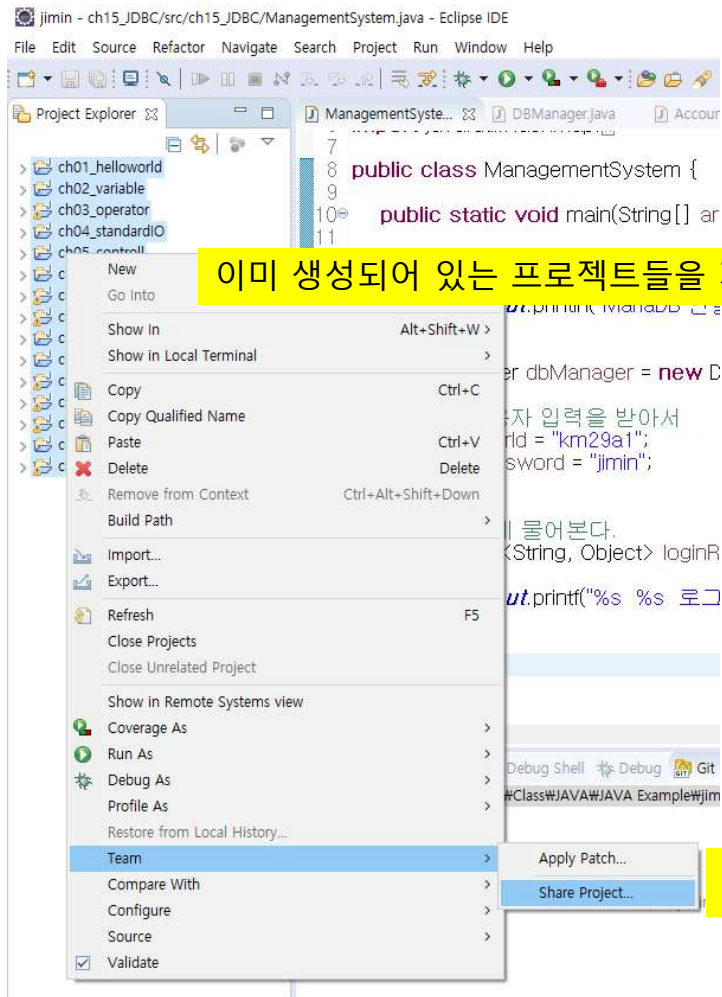


로컬에 저장할 위치 지정

이클립스에 git 붙이기 성공



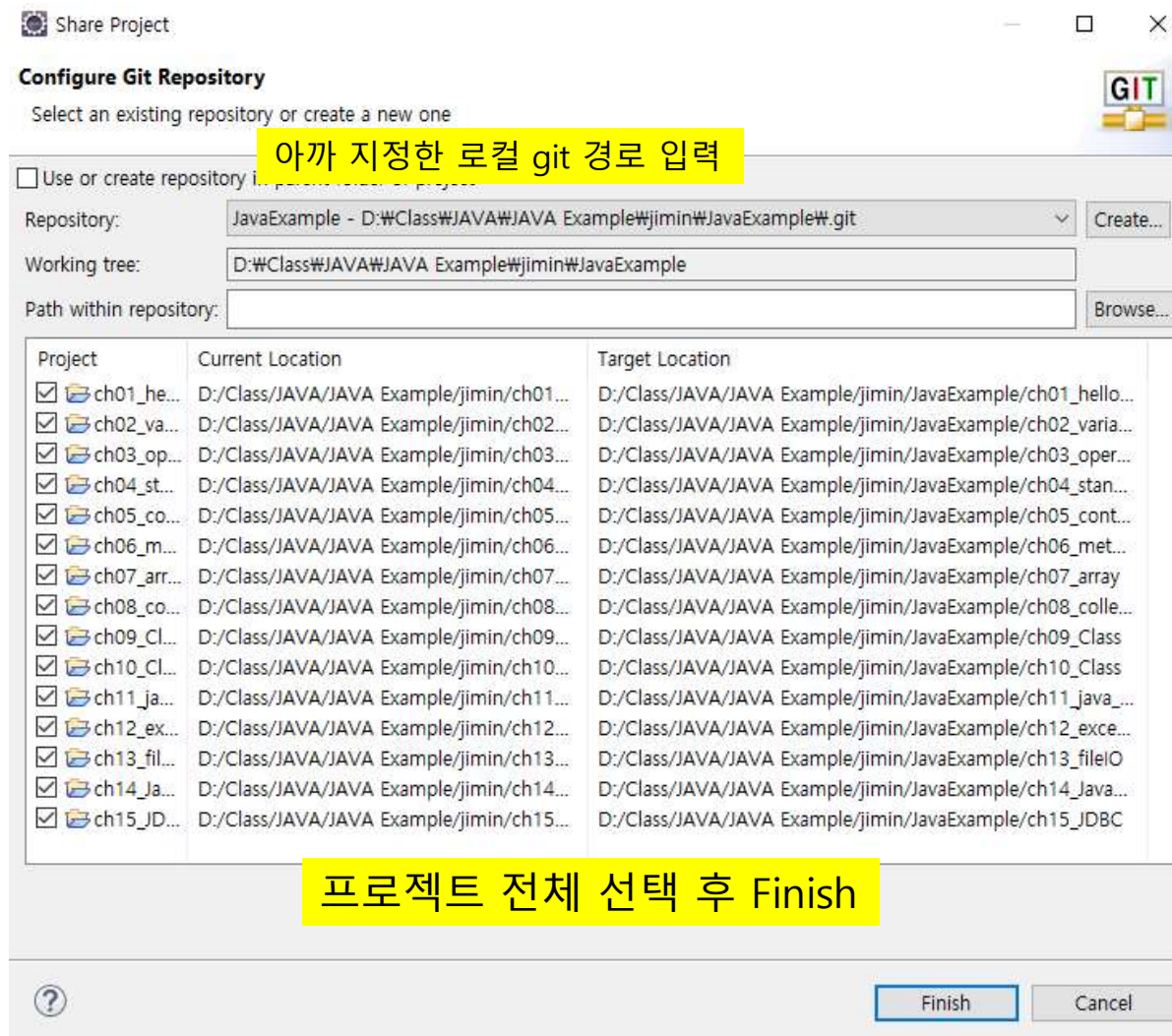
Git에다 만들었던 프로젝트 업로드하기



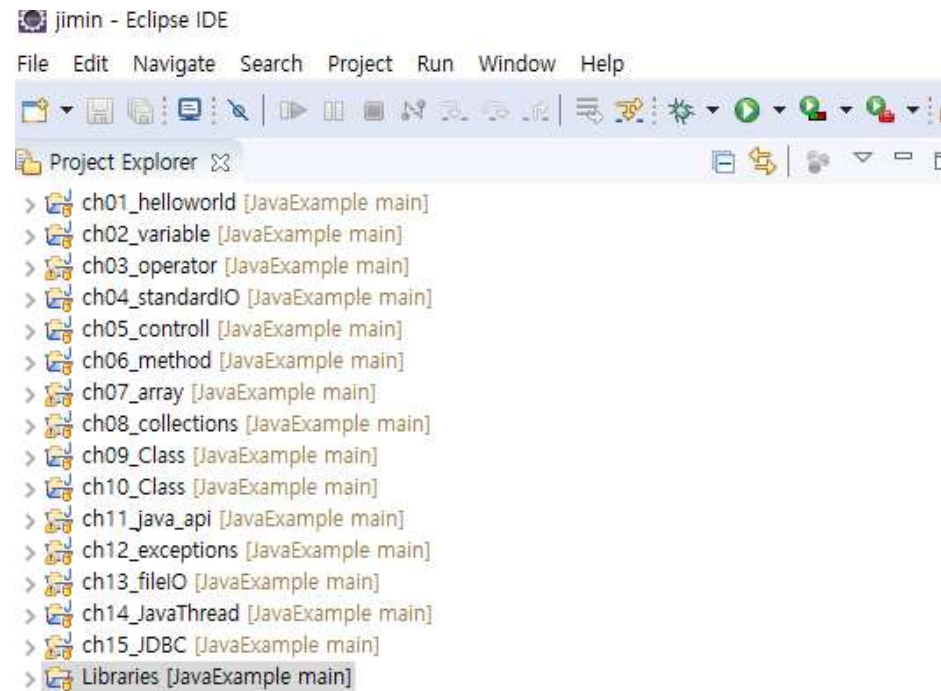
이미 생성되어 있는 프로젝트들을 지정 후 우클릭

우클릭 -> Team -> Share Project

Git에다 만들었던 프로젝트 업로드하기

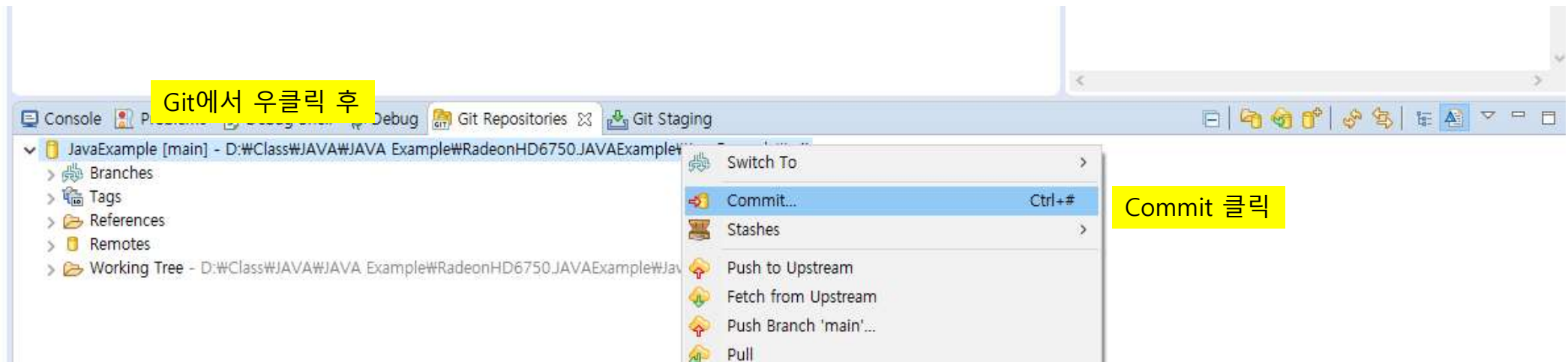


Git에다 만들었던 프로젝트 업로드하기

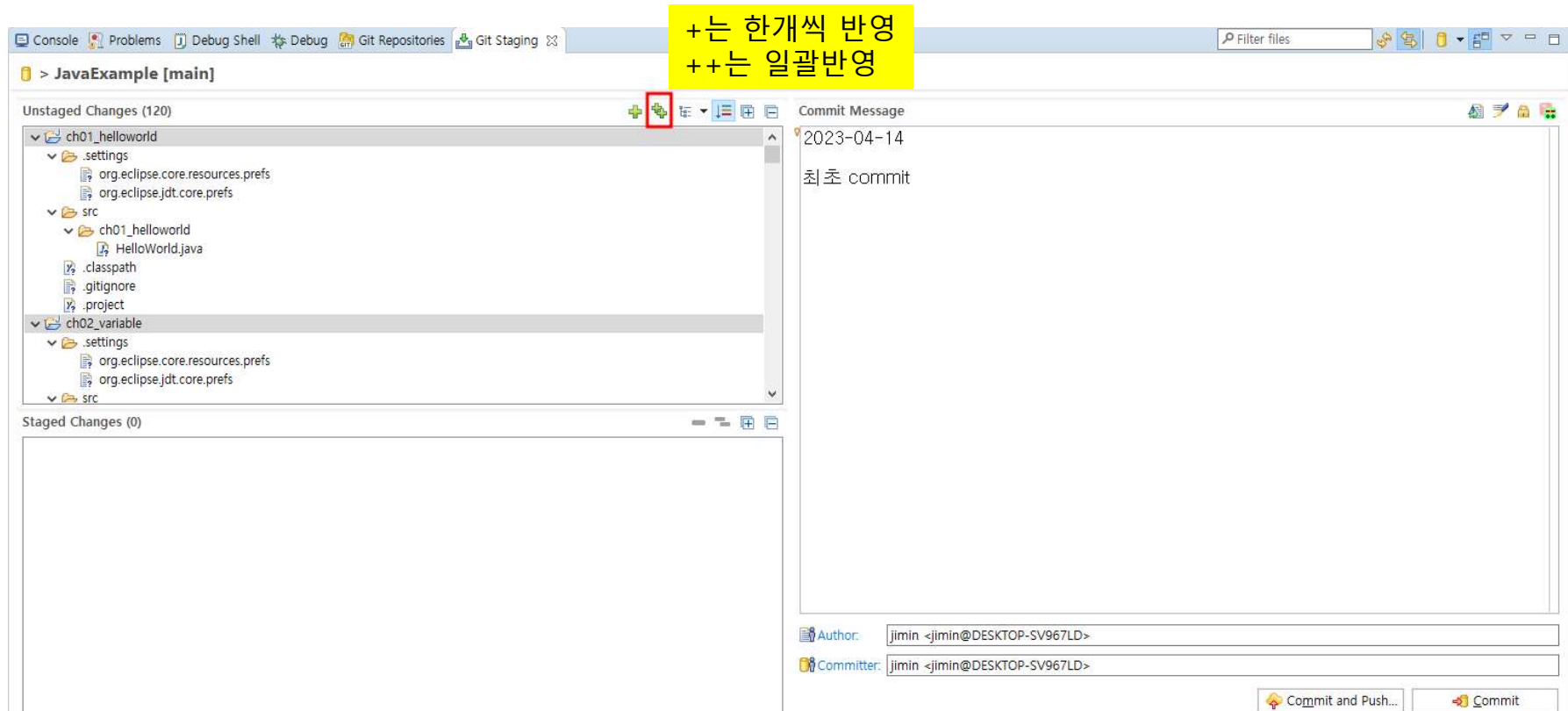


프로젝트명 옆에 git 이름이 나타나면 성공

Git에다 만들었던 프로젝트 업로드하기



Git에다 만들었던 프로젝트 업로드하기



Git에다 만들었던 프로젝트 업로드하기

The screenshot shows the Eclipse IDE interface with the Git Staging view active. The top toolbar includes buttons for Console, Problems, Debug Shell, Debug, Git Repositories, and Git Staging. The main area is divided into two panes: 'Unstaged Changes (0)' and 'Staged Changes (120)'. The 'Staged Changes' pane shows a file tree with folders like '.settings', 'src', and 'ch01_helloworld'. The 'Commit Message' pane on the right contains the text '2023-04-14' and '최초 commit'. Below the message pane, the 'Author' and 'Committer' fields are both set to 'jimin <jimin@DESKTOP-SV967LD>'. A yellow highlight box is placed over the 'Commit and Push' button in the bottom right corner.

Unstaged Changes (0)

Staged Changes (120)

Commit Message

2023-04-14
최초 commit

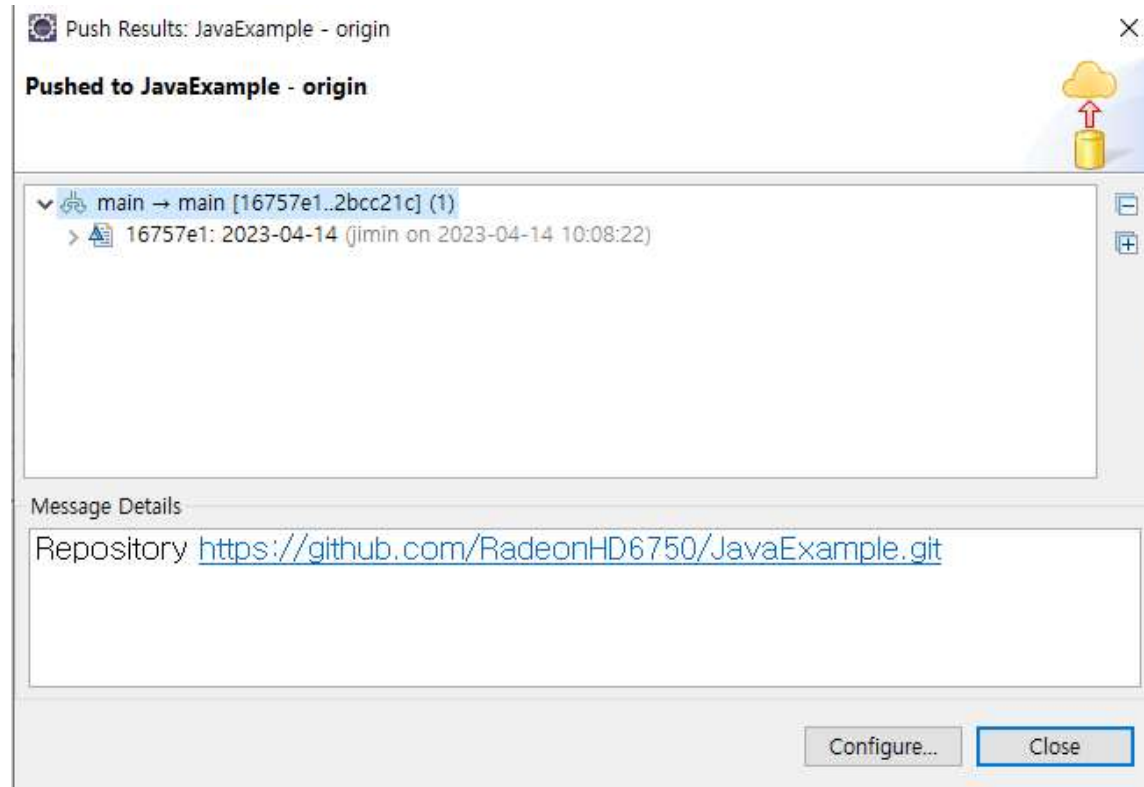
Author: jimin <jimin@DESKTOP-SV967LD>
Committer: jimin <jimin@DESKTOP-SV967LD>

Commit and Push

Commit and Push...

그냥 Commit만 하면 로컬 저장소에만
저장되고, 원격에는 안된다.

Git에다 만들었던 프로젝트 업로드하기



이렇게 뜨면 성공

Git에다 만들었던 프로젝트 업로드하기

[Pull requests](#)
[Issues](#)
[Codespaces](#)
[Marketplace](#)
[Explore](#)

[RadeonHD6750 / JavaExample](#)
Public

[Pin](#)
[Unwatch 1](#)
[Fork 0](#)
[Star 0](#)

[Code](#)
[Issues](#)
[Pull requests](#)
[Actions](#)
[Projects](#)
[Wiki](#)
[Security](#)
[Insights](#)
[Settings](#)

main
1 branch
0 tags

[Go to file](#)
[Add file](#)
[Code](#)

RadeonHD6750 Update README.md
d45efbd 1 hour ago
5 commits

Libraries	라이브러리 추가	1 hour ago
ch01_helloworld	2023-04-14	1 hour ago
ch02_variable	2023-04-14	1 hour ago
ch03_operator	2023-04-14	1 hour ago
ch04_standardIO	2023-04-14	1 hour ago
ch05_controll	2023-04-14	1 hour ago
ch06_method	2023-04-14	1 hour ago
ch07_array	2023-04-14	1 hour ago
ch08_collections	2023-04-14	1 hour ago
ch09_Class	2023-04-14	1 hour ago
ch10_Class	2023-04-14	1 hour ago
ch11_java_api	2023-04-14	1 hour ago

About

JAVA 수업 예제

Readme

GPL-3.0 license

0 stars

1 watching

0 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Languages

온라인 깃헙으로 다시 돌아가보면 반영이 되어 있을 것이다.

주의사항

강제이주

Left Screenshot (jimin):

이름	수정한 날짜	유형	크기
.metadata	2023-04-06 오전 8:41	파일 폴더	
commons-lang3-3.12.0.jar	2020-01-22 오후 3:10	압축(JAR) 파일	574KB
ini4j-0.5.4.jar	2015-02-17 오후 4:42	압축(JAR) 파일	100KB
ini4j-0.5.4-jdk14.jar	2015-02-17 오후 4:42	압축(JAR) 파일	104KB
json-simple-1.1.1.jar	2023-04-12 오전 10:47	압축(JAR) 파일	24KB
mysql-connector-java-8.0.29.jar	2023-04-13 오전 11:16	압축(JAR) 파일	624KB
ojdbc6-11.2.0.4.jar	2023-04-13 오전 9:46	압축(JAR) 파일	2,676KB
opencsv-5.7.1.jar	2023-04-10 오후 5:59	압축(JAR) 파일	234KB

Right Screenshot (RadeonHD6750.JAVAExample):

이름	수정한 날짜	유형	크기
.git	2023-04-14 오전 10:14	파일 폴더	
ch01_helloworld	2023-04-14 오전 10:08	파일 폴더	
ch02_variable	2023-04-14 오전 10:08	파일 폴더	
ch03_operator	2023-04-14 오전 10:08	파일 폴더	
ch04_standardIO	2023-04-14 오전 10:08	파일 폴더	
ch05_controll	2023-04-14 오전 10:08	파일 폴더	
ch06_method	2023-04-14 오전 10:08	파일 폴더	
ch07_array	2023-04-14 오전 10:08	파일 폴더	
ch08_collections	2023-04-14 오전 10:08	파일 폴더	
ch11_java	2023-04-14 오전 10:08	파일 폴더	
ch12_exceptions	2023-04-14 오전 10:08	파일 폴더	
ch13_fileIO	2023-04-14 오전 10:08	파일 폴더	
ch14_JavaThread	2023-04-14 오전 10:08	파일 폴더	
ch15_JDBC	2023-04-14 오전 10:08	파일 폴더	
Libraries	2023-04-14 오전 10:12	파일 폴더	
LICENSE	2023-04-14 오전 10:06	파일	35KB
README.md	2023-04-14 오전 10:06	MD 파일	1KB

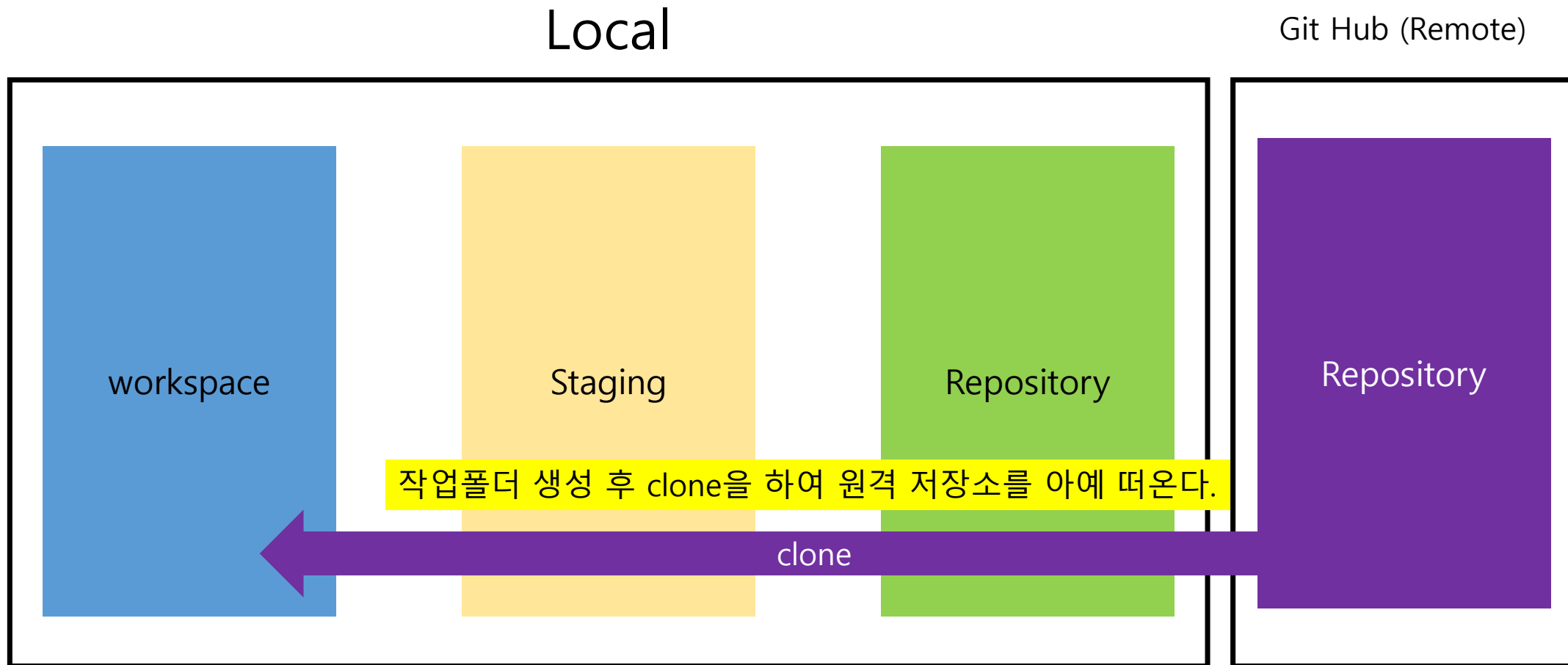
원래 있던 프로젝트 폴더가 통째로 새로 지정한 저장소로 강제 이주됨.

고로 처음부터 로컬 저장소를 만들고 거기서 프로젝트를 만들어야 한다.

GitHub으로 형상관리



최초 repository 작업



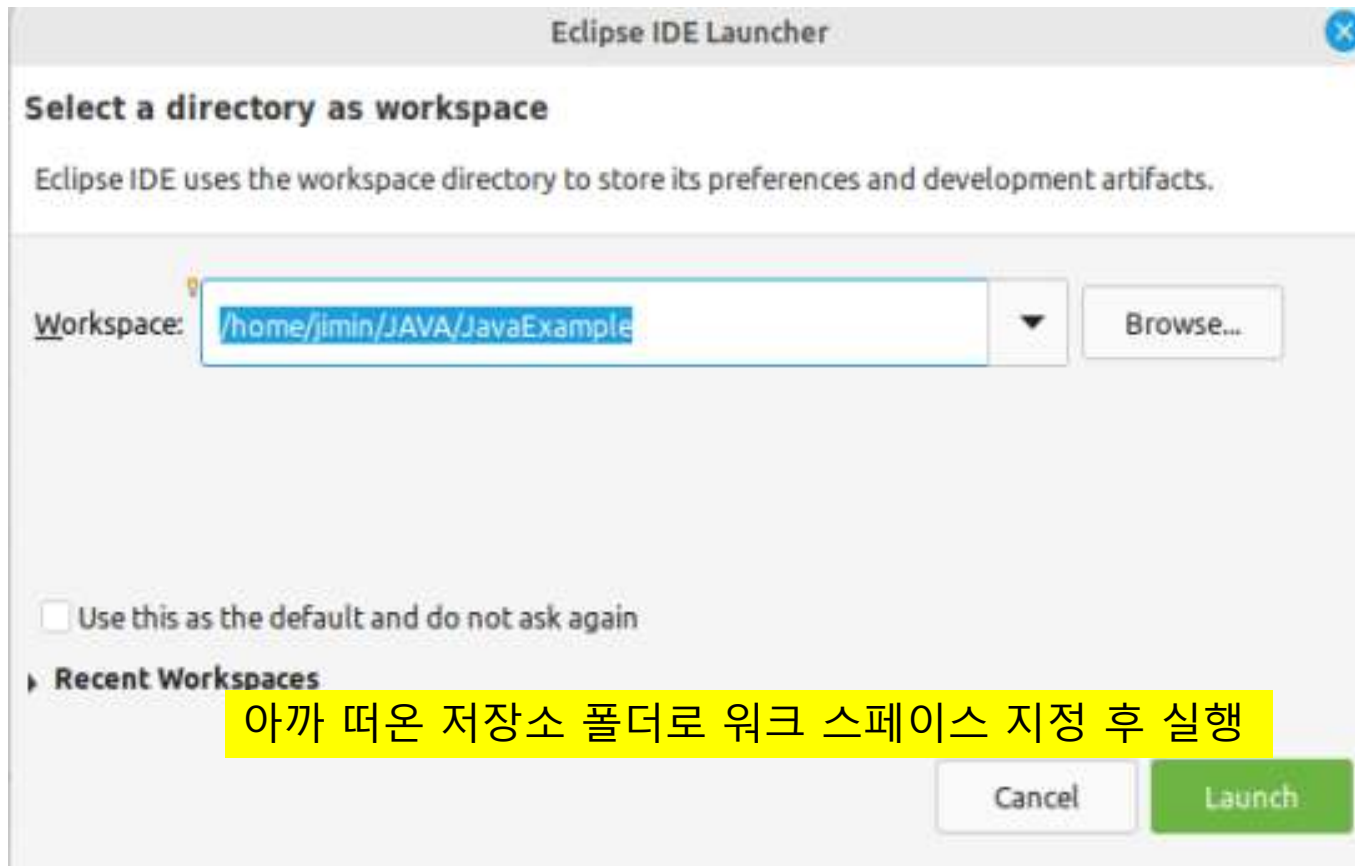
최초 repository 작업 (예시)

```
jimin@jimin-meat: ~/JAVA
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
jimin@jimin-meat:~/JAVA$ git clone https://github.com/RadeonHD6750/JavaExample.git
'JavaExample'에 복제합니다...
remote: Enumerating objects: 189, done.
remote: Counting objects: 100% (189/189), done.
remote: Compressing objects: 100% (105/105), done.
remote: Total 189 (delta 64), reused 163 (delta 49), pack-reused 0
오브젝트를 받는 중: 100% (189/189), 4.05 MiB | 21.07 MiB/s, 완료.
델타를 알아내는 중: 100% (64/64), 완료.
jimin@jimin-meat:~/JAVA$
```

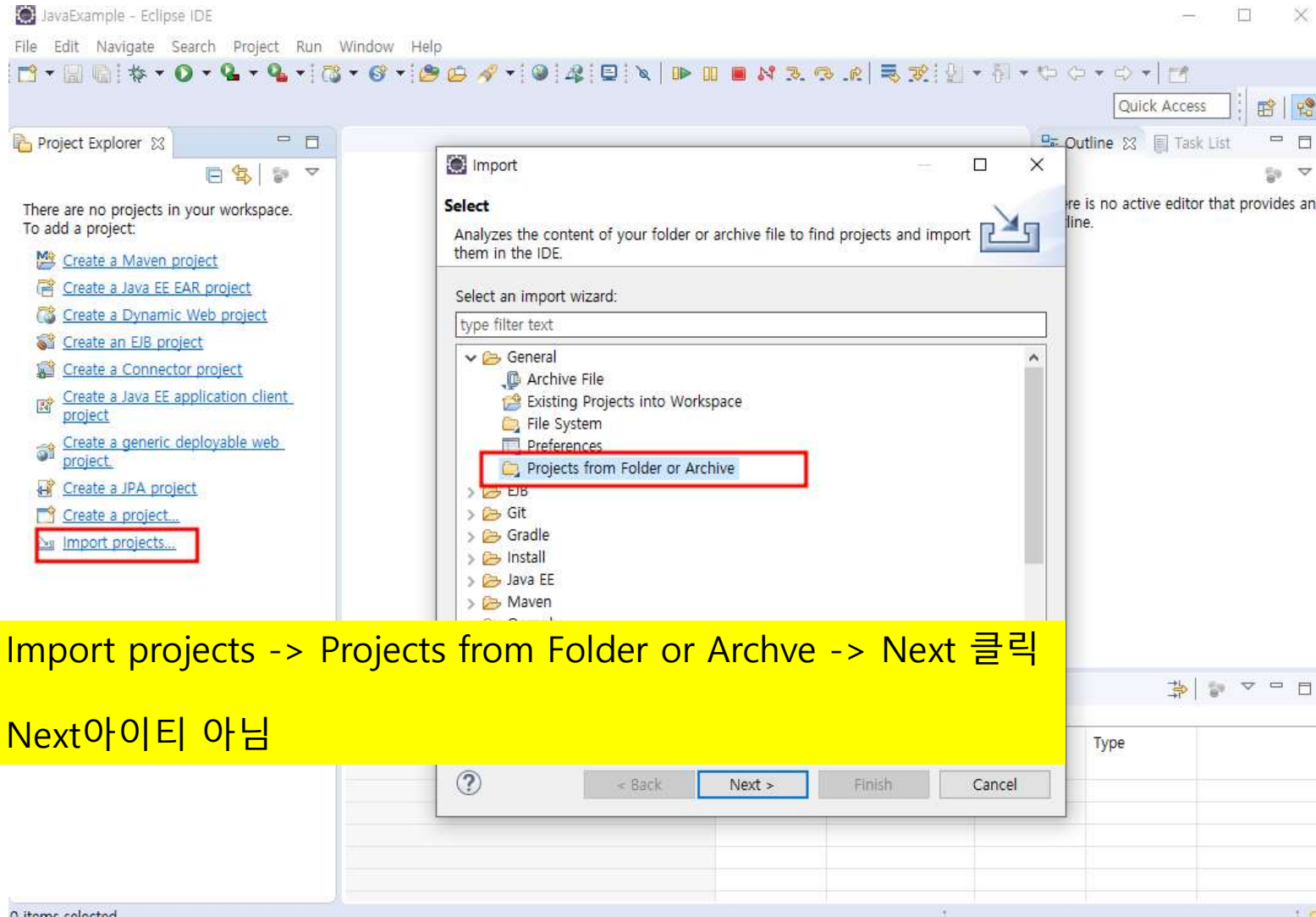
깃헙에 있는 저장소 다운로드 완료

git clone "주소"를 입력하여 저장소 다운로드

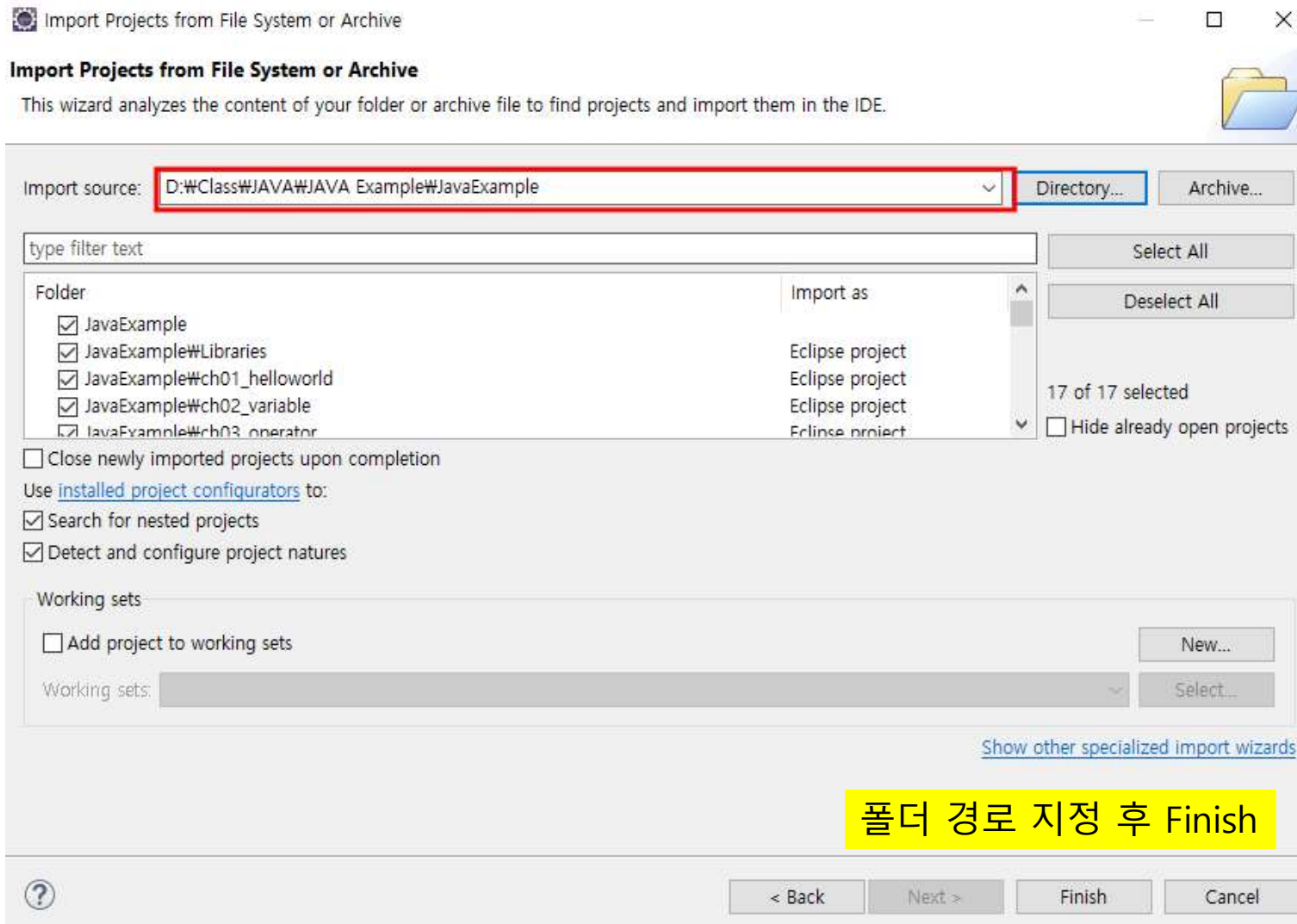
최초 repository 작업 (예시)



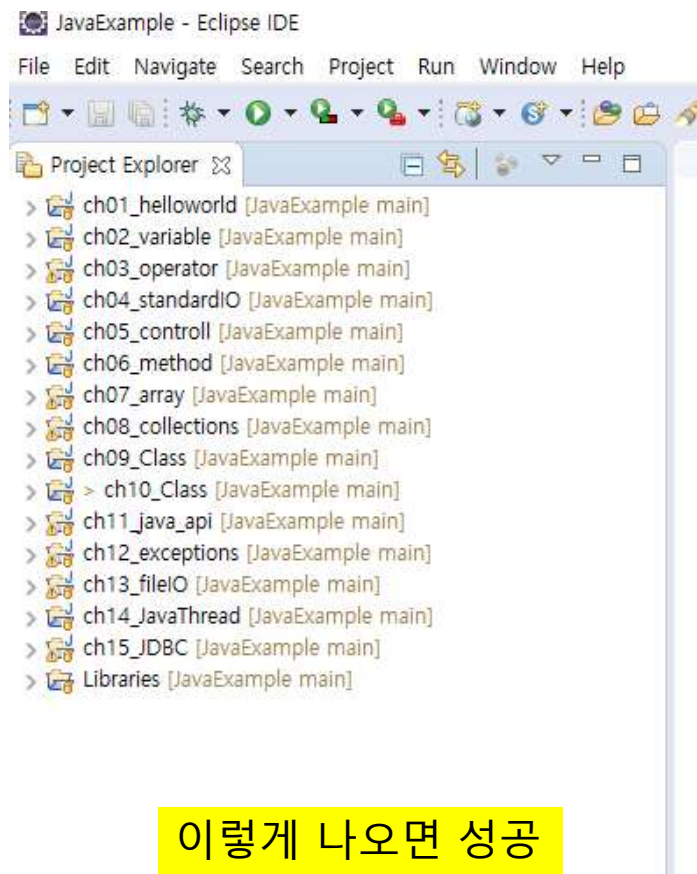
최초 repository 작업 (예시)



최초 repository 작업 (예시)



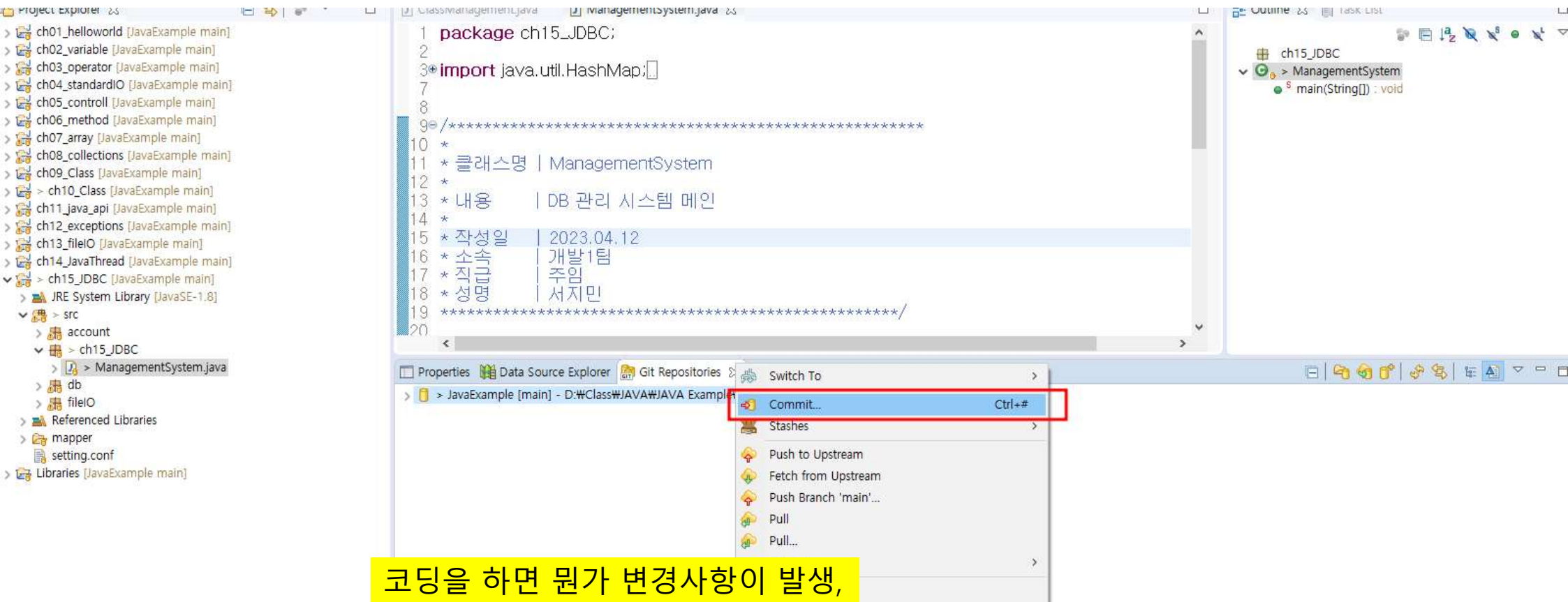
최초 repository 작업 (예시)



이렇게 나오면 성공

이제 언제 어디서든 깃허브 저장소에 있는 소스들을 가져올 수 있다.
또한, 강제이주를 막을 수 있다.

Git 업로드 작업 (예시)

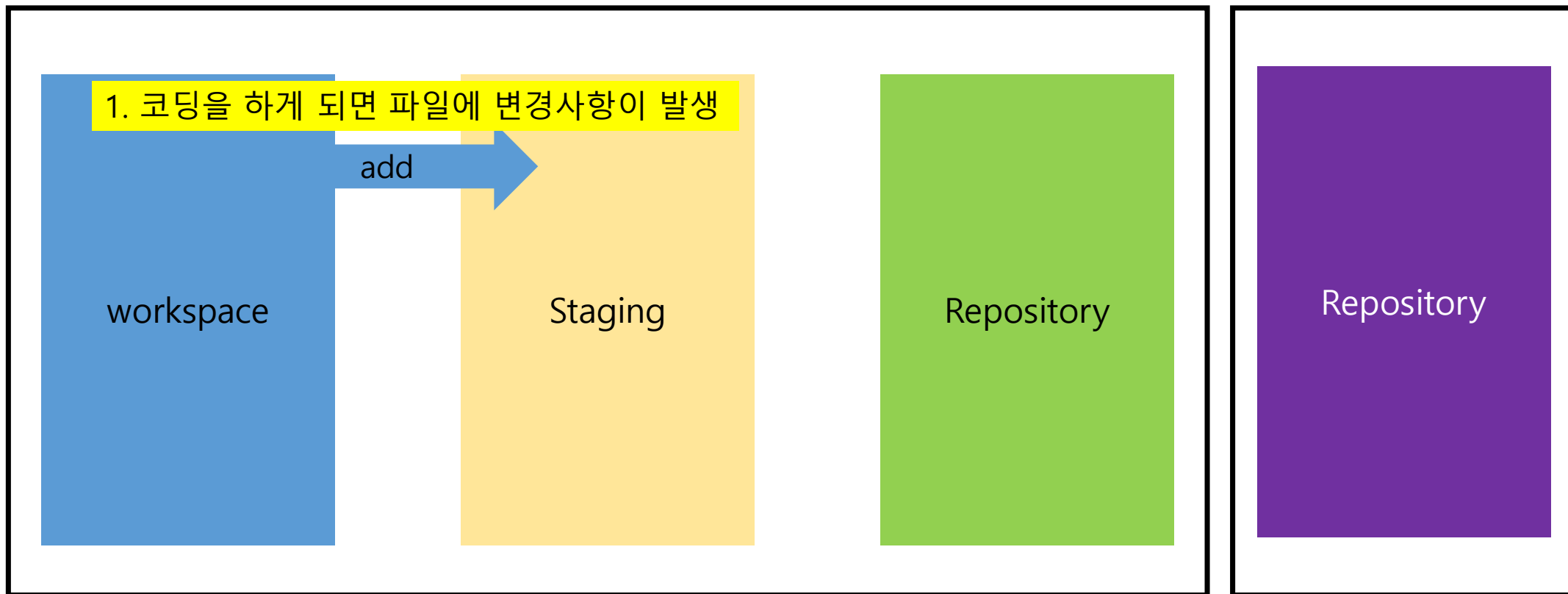


코딩을 하면 뭔가 변경사항이 발생,
이후 Commit 버튼 클릭

Git 업로드 작업순서

Local

Git Hub (Remote)



Git 업로드 작업 (예시) – add 과정

Properties Data Source Explorer Git Repositories Git Staging Filter files

> JavaExample [main] Staging탭에서 최종 확인

Unstaged Changes (9)

- .gitignore
- .gitignore - ch10_Class/bin
- ClassManagement.class - ch10_Class/bin/ch09_Class
- ClassRoom.class - ch10_Class/bin/ch09_Class
- > ManagementSystem.java - ch15_JDBC/src/ch15_JDBC
- Person.class - ch10_Class/bin/ch10_person
- Student.class - ch10_Class/bin/ch10_person
- Teacher.class - ch10_Class/bin/ch10_person
- Util.class - ch10_Class/bin/ch09_Functions

Staged Changes (0)

Commit Message

Author: jimin <jimin@DESKTOP-SV967LD>

Committer: jimin <jimin@DESKTOP-SV967LD>

Commit and Push... Commit

Git 업로드 작업 (예시) – add 과정

Properties Data Source Explorer Git Repositories Git Staging Filter files

> JavaExample [main]

Unstaged Changes (0)

Commit Message

메시지는 팀원이 알아볼 수 있게
친절히 작성

Staged Changes (9)

- .gitignore
- .gitignore - ch10_Class/bin
- ClassManagement.class - ch10_Class/bin/ch09_Class
- ClassRoom.class - ch10_Class/bin/ch09_Class
- ManagementSystem.java - ch15_JDBC/src/ch15_JDBC
- Person.class - ch10_Class/bin/ch10_person
- Student.class - ch10_Class/bin/ch10_person
- Teacher.class - ch10_Class/bin/ch10_person
- Util.class - ch10_Class/bin/ch09_Functions

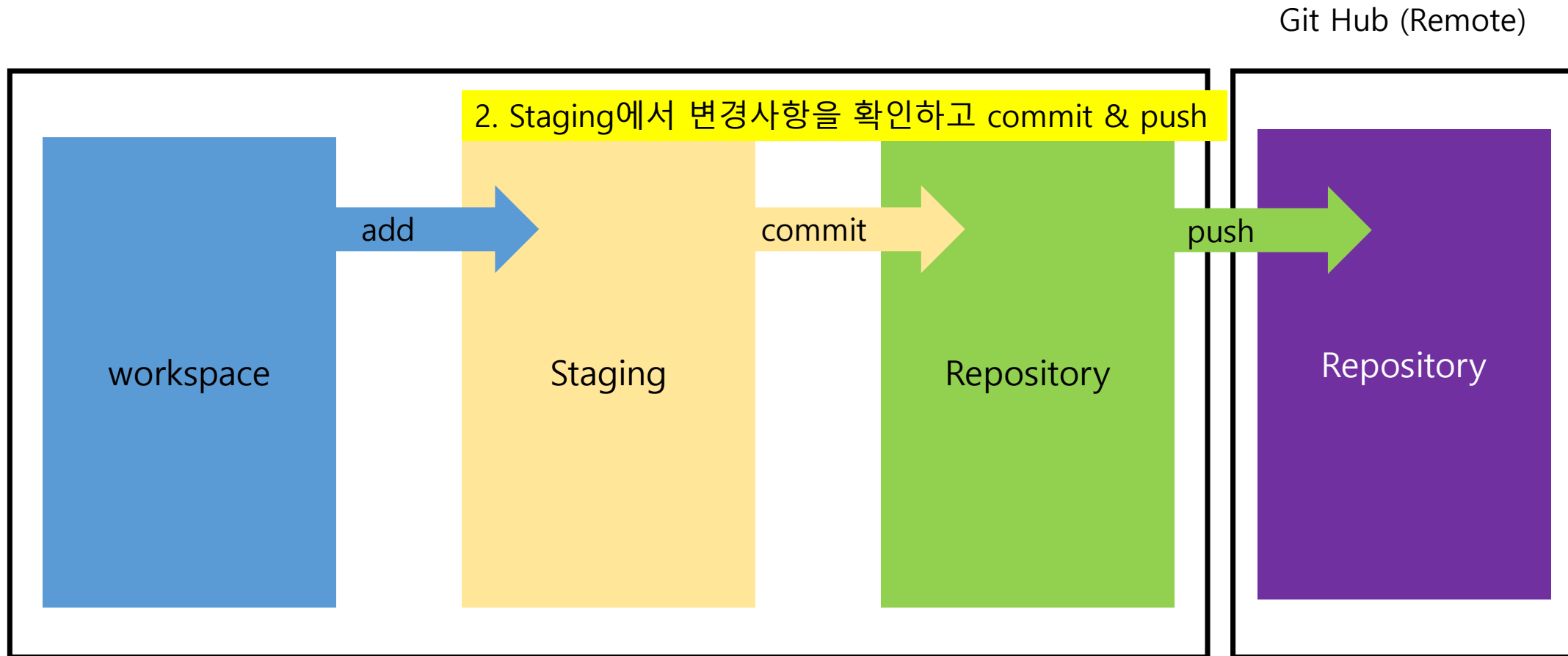
Author: jimin <jimin@DESKTOP-SV967LD>

Committer: jimin <jimin@DESKTOP-SV967LD>

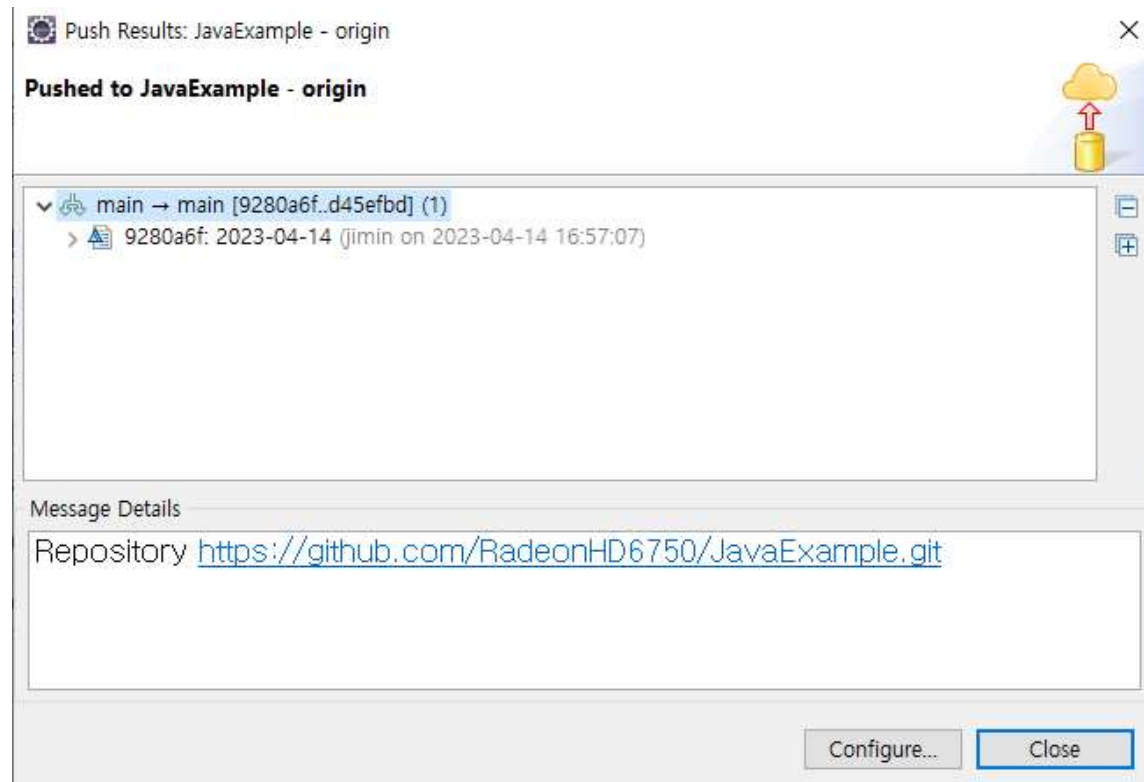
Staging탭에서 최종 확인 후 Commit and Push=>

Commit and Push... Commit

Git 업로드 작업순서



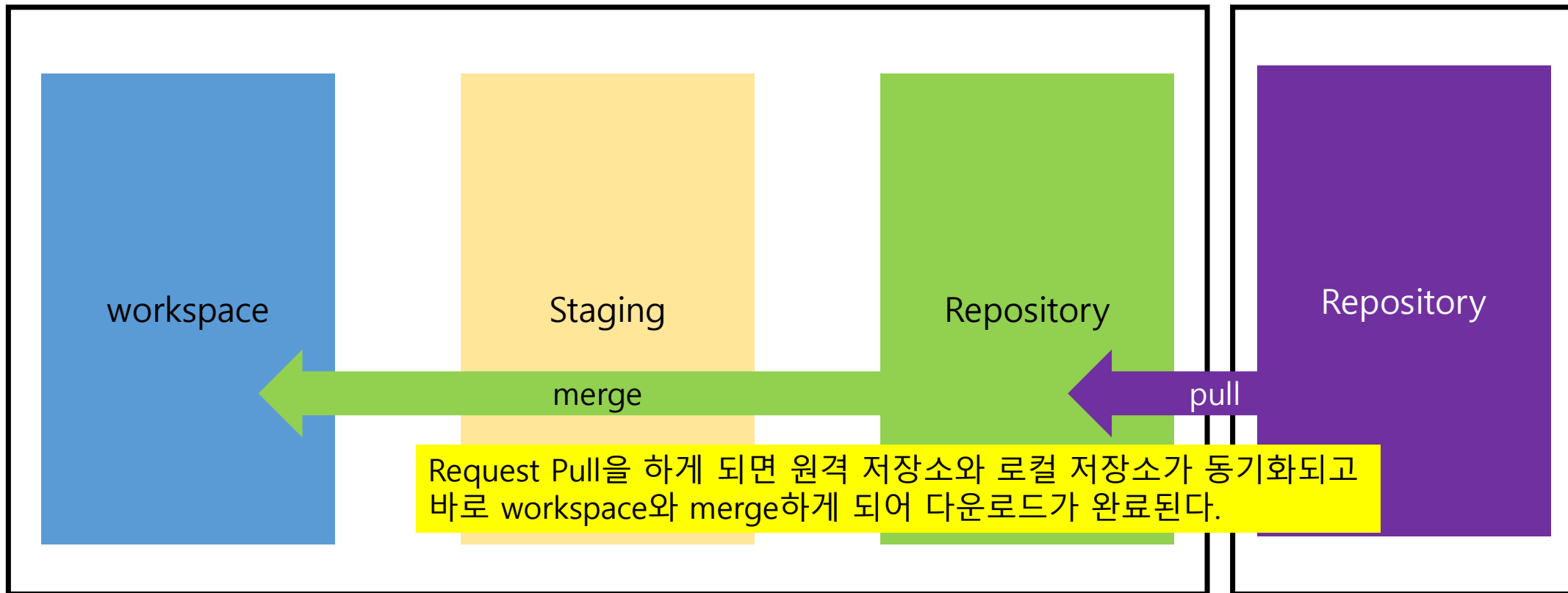
Git 업로드 작업 (예시)



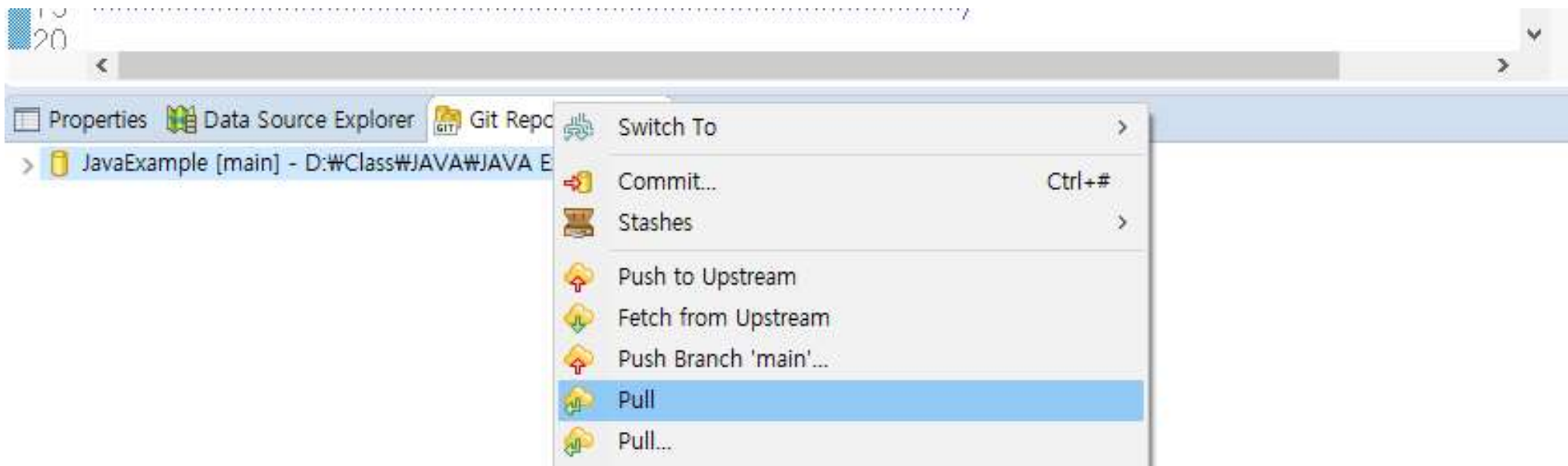
이렇게 나오면 성공

Git 코드 동기화 작업순서

팀원이 작성한 코드를 내려 받아 동기화할 때 사용하며,
출근 하면 제일 먼저 해야 될 일중 하나이다.

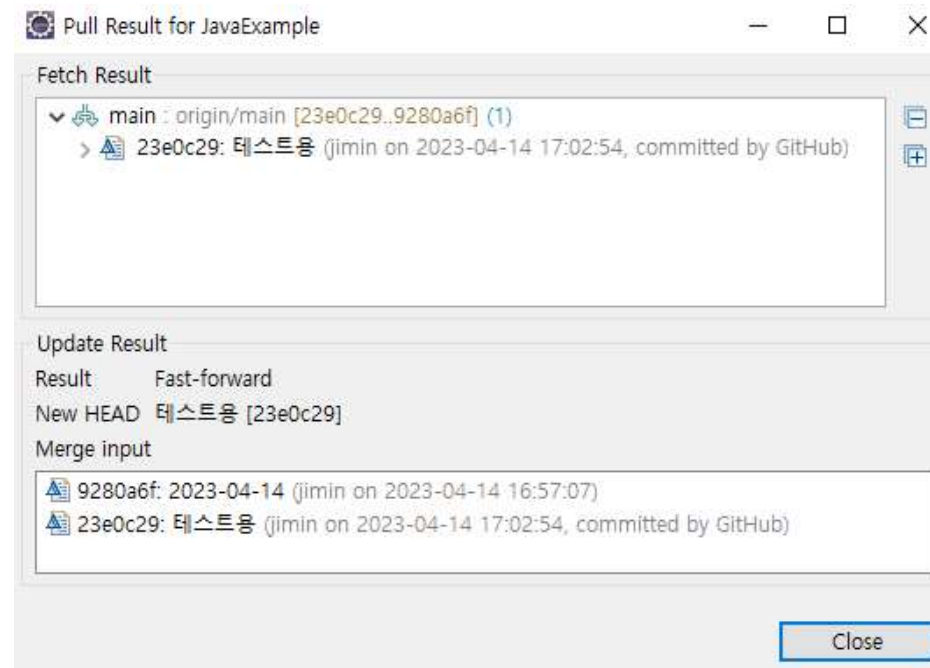


Git 코드 동기화 작업 (예시)



Pull 동작 '첫번째'꺼 선택

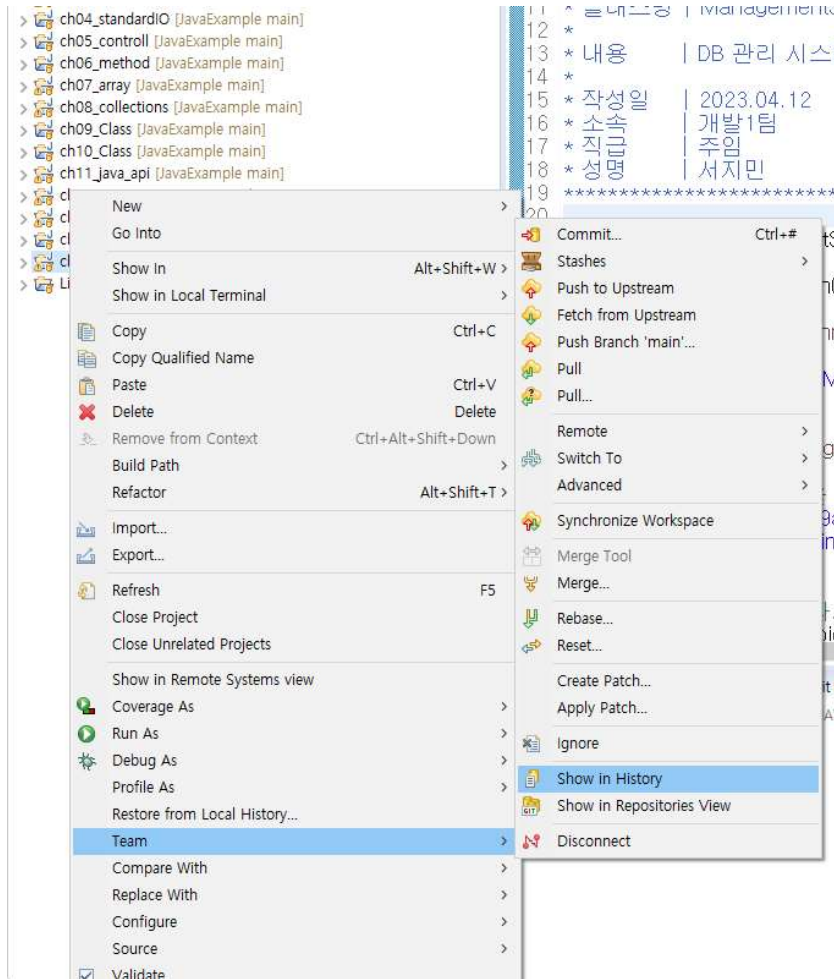
Git 코드 동기화 작업 (예시)



Pull 동작 이후 이렇게 나오면 성공

자동으로 local repo와 workspace가 merge된다

Github 고급 사용법1 - 히스토리



협업시 거진 항상 사용하는 기능이다. 아무리 잘 의 사소통하여도 한번씩은 으레 소스코드 충돌이 발생한다.

이 때 분쟁조정시 주로 사용한다. 따라서 출근하면 항상 첫번째로 해야 될 일이 pull을 하여서 동기화 해 주는 작업이다. 물론 이렇게 하여도 으레 소스코드 충돌이 발생한다.

프로젝트 우클릭 -> Team -> show in history 클릭

Github 고급 사용법1 - 히스토리

Properties

Data Source Explorer

Git Repositories

Git Staging

History

Project: ch15_JDBC [JavaExample]

Id	Message	Author	Authored Date	Committer	Committed Date
23e0c29	main origin/HEAD origin/main HEAD 테스트용	jimin	3 days ago	GitHub	3 days ago
9280a6f	2023-04-14	jimin	3 days ago	jimin	3 days ago
af6b189	경로수정	jimin	3 days ago	jimin	3 days ago
16757e1	2023-04-14	jimin	3 days ago	jimin	3 days ago

Commit 내역

commit 9280a6fa838f6a193f1faa4b751ba67e2d388042

Author: jimin <jimin@DESKTOP-SV967LD> 2023-04-14 16:57:07

Committer: jimin <jimin@DESKTOP-SV967LD> 2023-04-14 16:57:07

Parent: af6b189a9d3f76aa11464e3983ac10c59da92f82 (경로수정)

Child: 23e0c290a16e71a5c960402892f2b66d6acc0998 (테스트용)

Branches: [main](#), [origin/HEAD](#), [origin/main](#)

2023-04-14

commit 테스트

Commit 상세정보

gitignore

ch10_Class/bin/.gitignore

ch10_Class/bin/ch09_Class/ClassManagement.class

ch10_Class/bin/ch09_Class/ClassRoom.class

ch10_Class/bin/ch09_Functions/Util.class

ch10_Class/bin/ch10_person/Person.class

ch10_Class/bin/ch10_person/Student.class

ch10_Class/bin/ch10_person/Teacher.class

ch15_JDBC/src/ch15_JDBC/ManagementSystem.java

수정된 파일

Github 고급 사용법1 - 히스토리

Java Source Compare ▾

ManagementSystem.java 9280a6f (jimin)

ManagementSystem.java d45efbd (jimin)

```

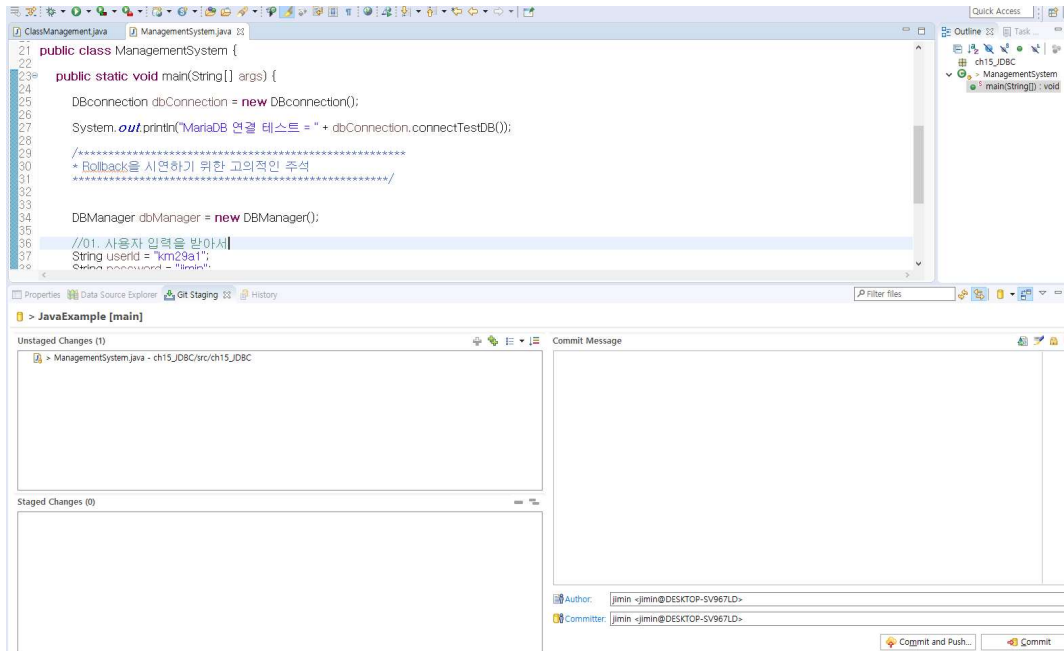
1 package ch15_JDBC;
2
3 import java.util.HashMap;
4
5 import db.DBManager;
6 import db.DBconnection;
7
8
9 /*****
10 *
11 * 클래스명 | ManagementSystem
12 *
13 * 내용 | DB 관리 시스템 메인
14 *
15 * 작성일 | 2023.04.12
16 * 소속 | 개발1팀
17 * 직급 | 주임
18 * 성명 | 서지민
19 *****/
20
21 public class ManagementSystem {
22
23     public static void main(String[] args) {
24
25         DBconnection dbConnection = new DBconnection();
    
```

```

1 package ch15_JDBC;
2
3 import java.util.HashMap;
4
5 import db.DBManager;
6 import db.DBconnection;
7
8 public class ManagementSystem {
9
10     public static void main(String[] args) {
11
12         DBconnection dbConnection = new DBconnection();
13
14         System.out.println("MariaDB 연결 테스트 = " + dbConnection.conne
15
16
17         DBManager dbManager = new DBManager();
18
19         //01. 사용자 입력을 받아서
20         String userId = "km29a1";
21         String password = "jimin";
22
23
24         //02. DB에 물어본다.
25         HashMap<String, Object> loginResult = dbManager.loginTransaction
    
```

파일 선택시 나오는 상세 수정내용

Github 고급 사용법2 - 복구하기



git을 사용하는 어쩌면 궁극적인 목적이 바로 복구이다. Commit을 시행했던 특정시점으로 돌아가는 복구 기법이 사용된다.

Rollback 테스트를 위하여 주석 작성 및 commit & push

Github 고급 사용법2 - 복구하기

Java Source Compare

ManagementSystem.java f9f2abb (jimin)

```

23 public static void main(String[] args) {
24
25     DBConnection dbConnection = new DBConnection();
26
27     System.out.println("MariaDB 연결 테스트 = " + dbConnection.connectTest());
28
29     /* Rollback을 시연하기 위한 고의적인 주석 */
30     /* Rollback을 시연하기 위한 고의적인 주석 */
31
32
33     DBManager dbManager = new DBManager();
34
35     //01. 사용자 입력을 받아서
36

```

ManagementSystem.java 23e0c29 (jimin)

```

23 public static void main(String[] args) {
24
25     DBConnection dbConnection = new DBConnection();
26
27     System.out.println("MariaDB 연결 테스트 = " + dbConnection.connectTest());
28
29
30     DBManager dbManager = new DBManager();
31
32     //01. 사용자 입력을 받아서
33     String userId = "km29a1";
34     String password = "jimin";
35
36

```

id	Message	Author	Authored Date	Committer	Committed Date
f9f2abb	Rollback 시연용 Commit	jimin	9 seconds ago	jimin	9 seconds ago
23e0c29	테스트용	jimin	3 days ago	GitHub	3 days ago
9280a6f	2023-04-14	jimin	3 days ago	jimin	3 days ago
16757e1	2023-04-14	jimin	3 days ago	jimin	3 days ago

commit f9f2abb83a040815b9da133aee8a4593c5303af3
 Author: jimin <jimin@DESKTOP-SV967LD> 2023-04-17 17:28:59
 Committer: jimin <jimin@DESKTOP-SV967LD> 2023-04-17 17:28:59
 Parent: 23e0c290a16e71a5c960402892f2b66d6acc0998 (테스트용)
 Branches: main, origin/HEAD, origin/main

Rollback 시연용 Commit

```

diff --git a/ch15_JDBC/src/ch15_JDBC/ManagementSystem.java b/ch15_JDBC/src/ch15_JDBC/ManagementSystem.java
index a865591..4bb80ca 100644
--- a/ch15_JDBC/src/ch15_JDBC/ManagementSystem.java

```

테스트용 주석이 잘 commit 된 모습이다.

Github 고급 사용법2 - 복구하기

Reset – 롤백 이력 안남김

(현시점 복구불가)

Revert – 롤백 이력 남김

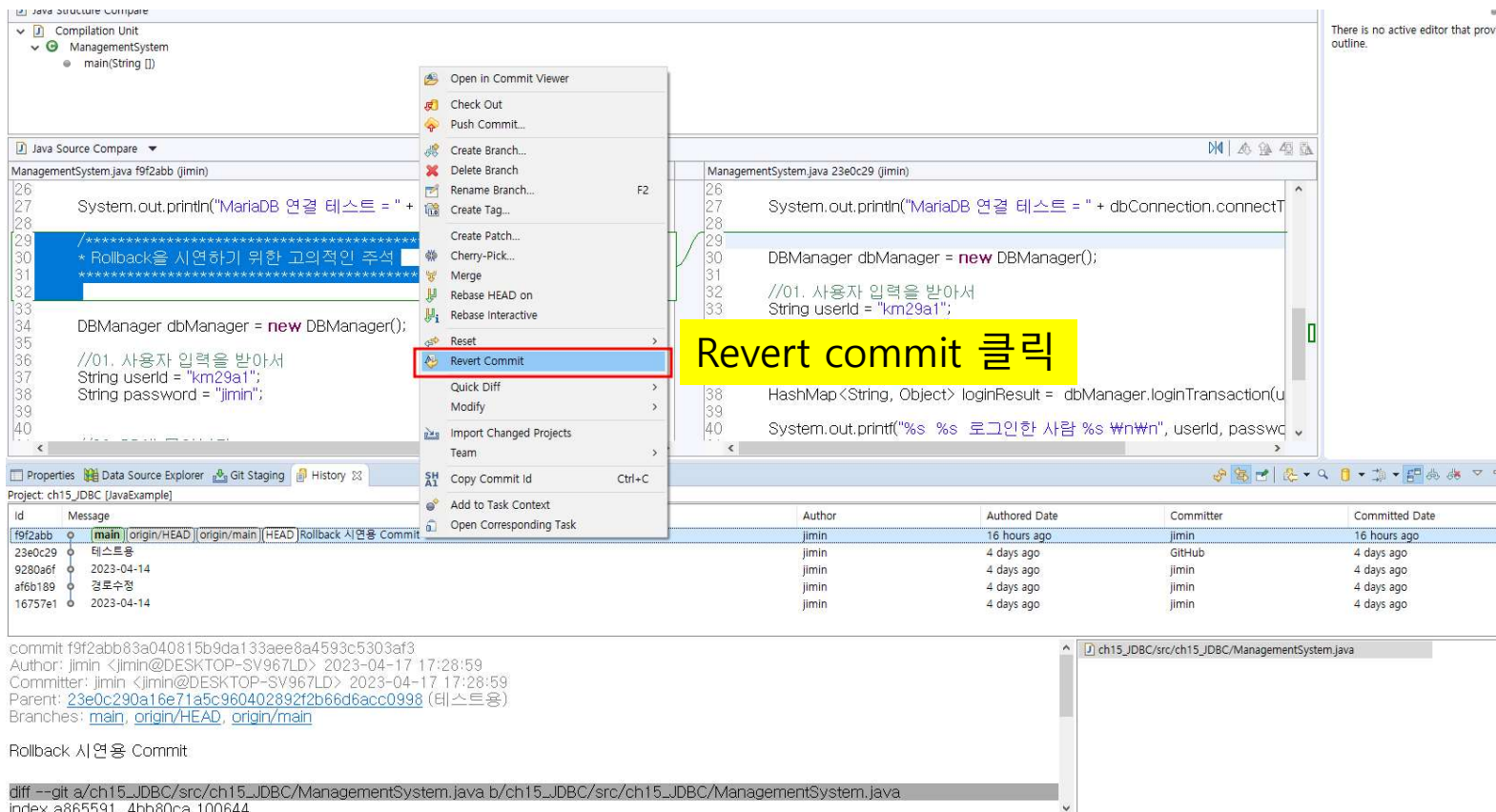
(롤백 취소 가능)

그냥 가만히 있을 걸...이라는

상황이 있을 수 있음으로

웬만해서는 롤백 이력을

남겨야 한다.



Revert commit 클릭

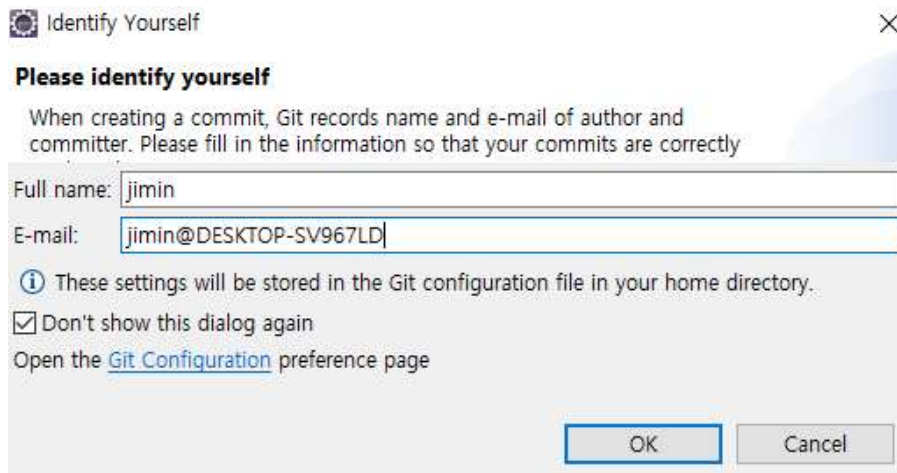
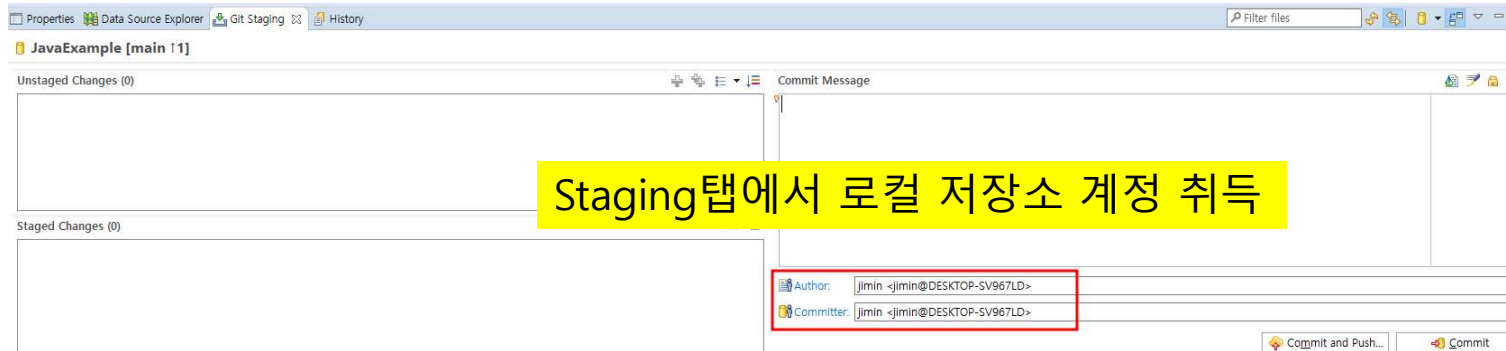
Author	Authored Date	Committer	Committed Date
jimin	16 hours ago	jimin	16 hours ago
jimin	4 days ago	GitHub	4 days ago
jimin	4 days ago	jimin	4 days ago
jimin	4 days ago	jimin	4 days ago
jimin	4 days ago	jimin	4 days ago

commit f9f2abb83a040815b9da133aee8a4593c5303af3
 Author: jimin <jimin@DESKTOP-SV967LD> 2023-04-17 17:28:59
 Committer: jimin <jimin@DESKTOP-SV967LD> 2023-04-17 17:28:59
 Parent: 23e0c290a16e71a5c960402892f2b66d6acc0998 (테스트용)
 Branches: main, origin/HEAD, origin/main

Rollback 시연용 Commit

diff --git a/ch15_JDBC/src/ch15_JDBC/ManagementSystem.java b/ch15_JDBC/src/ch15_JDBC/ManagementSystem.java
 index a865591..4bb80ca 100644

Github 고급 사용법2 - 복구하기



이상이 없다면 자동입력되어 있음

Github 고급 사용법2 - 복구하기

Java Source Compare

Local: ManagementSystem.java

```

23 public static void main(String[] args) {
24     DBConnection dbConnection = new DBConnection();
25     System.out.println("MariaDB 연결 테스트 = " + dbConnection.connectTestDB());
26
27     DBManager dbManager = new DBManager();
28
29     //01. 사용자 입력을 받아서
30     String userId = "km29a1";
31     String password = "jimin";
32
33     //02. DB에 물어본다.
34     HashMap<String, Object> loginResult = dbManager.loginTransaction(userId, password);
35
36
37
38

```

ManagementSystem.java f9f2abb (jimin)

```

23 public static void main(String[] args) {
24     DBConnection dbConnection = new DBConnection();
25     System.out.println("MariaDB 연결 테스트 = " + dbConnection.connectTestDB());
26
27     DBManager dbManager = new DBManager();
28
29     /*****
30      * Rollback을 시연하기 위한 고의적인 주석
31      *****/
32
33     DBManager dbManager = new DBManager();
34     String password = "jimin";
35
36
37
38

```

초록색 상자와는 달리 빨간색 상자로 되어 있다.

Id	Message	Author	Authored Date	Committer	Committed Date
2fb7879	[main] HEAD Revert "Rollback 시연용 Commit"	jimin	3 minutes ago	jimin	3 minutes ago
f9f2abb	[origin/HEAD] Rollback 시연용 Commit	jimin	16 hours ago	jimin	16 hours ago
23e0c29	테스트용	jimin	4 days ago	GitHub	4 days ago
9280a6f	2023-04-14	jimin	4 days ago	jimin	4 days ago
16757e1	2023-04-14	jimin	4 days ago	jimin	4 days ago

commit: 2fb7879ef967edcc2b22a5ca5c2ad3b171b125c
 Author: jimin <jimin@DESKTOP-SV967LD> 2023-04-18 09:09:16
 Committer: jimin <jimin@DESKTOP-SV967LD> 2023-04-18 09:09:16
 Parent: f9f2abb83a040815b9da133aee8a4593c5303af3 (Rollback 시연용 Commit)
 Branches: main

Revert "Rollback 시연용 Commit"

This reverts commit f9f2abb83a040815b9da133aee8a4593c5303af3.

롤백한 이력이 남아 있다.

Github 고급 사용법2 - 복구하기

The screenshot shows an IDE with a Java file named `ManagementSystem.java`. The code includes a class `ManagementSystem` with a `main` method. The `main` method contains logic for database connection and login. The code is as follows:

```

11 * 클래스명 | ManagementSystem
12 *
13 * 내용 | DB 관리 시스템 메인 222
14 *
15 * 작성일 | 2023.04.12
16 * 소속 | 개발1팀
17 * 직급 | 주임
18 * 성명 | 서지민
19 *****/
20
21 public class ManagementSystem {
22
23     public static void main(String[] args) {
24
25         DBConnection dbConnection = new DBConnection();
26
27         System.out.println("MariaDB 연결 테스트 = " + dbConnection.connectTestDB());
28
29
30         DBManager dbManager = new DBManager();
31
32         //01. 사용자 입력을 받아서
33         String userId = "km29a1";
34         String password = "jimin";
35
36
37         //02. DB에 물어본다.
38         HashMap<String, Object> loginResult = dbManager.loginTransaction(userId, password);
39
40         System.out.printf("%s %s 로그인한 사람 %s\n\n", userId, password, loginResult.get("name"));
41
42     }
43 }

```

Below the code editor, the Git history is displayed. The table shows the following commits:

Id	Message	Author	Authored Date	Committer	Committed Date
2fb7879	(main) (HEAD) Revert "Rollback 시연용 Commit"	jimin	3 minutes ago	jimin	3 minutes ago
f9f2abb	(origin/HEAD) (origin/main) Rollback 시연용 Commit	jimin	16 hours ago	jimin	16 hours ago
23e0c29	테스트용	jimin	4 days ago	GitHub	4 days ago
0200e6f	2023.04.14	jimin	4 days ago	jimin	4 days ago

The commit message for the selected commit (2fb7879) is: "Rollback 시연용 Commit". The commit details show the author as jimin, the committer as jimin, and the parent commit as f9f2abb83a040815b9da133aee8a4593c5303af3 (Rollback 시연용 Commit).

주석을 작성하기 이전으로 돌아가 있다.

Github 고급 사용법2 - 복구하기

```

/

```

```

public class ManagementSystem {

```

```

    public static void main(String[] args) {

```

```

        DBConnection dbConnection = new DBConnection();

```

```

        System.out.println("MariaDB 연결 테스트 = " + dbConnection.connectTestDB());

```

```

        /*****
        * Rollback을 시연하기 위한 고의적인 주석
        *****/

```

```

        DBManager dbManager = new DBManager();

```

```

        //01. 사용자 입력을 받아서

```

```

        String userId = "km29a1";

```

```

        String password = "jimin";

```

```

        //02. DB에 물어본다.

```

```

        HashMap<String, Object> loginResult = dbManager.loginTransaction(userId, password);

```

```

        System.out.printf("%s %s 로그인한 사람 %s \n\n", userId, password, loginResult.get("name"));

```

```

    }

```

원격 저장소인 GitHub에는 해당 주석이 남아 있다.

이때, **롤백된 로컬저장소에서 충분한 테스트를** 진행하면 된다.

롤백한 것이 옳았다면 이제 롤백한 commit을 push 해주자


Github 고급 사용법2 - 복구하기


entityManager.java 2107079 and 1912a00 - Eclipse IDE

최종 Push commit

Id	Message	Author	Authored Date	Committer
2fb7879	main HEAD Revert "Rollback 시연용 Commit"	jimin	8 hours ago	jimin
f9f2abb	origin/HEAD (origin/main) Rollback 시연용 Commit	jimin	24 hours ago	jimin
23e0c29	테스트용	jimin	4 days ago	GitHub
9280a6f	2023-04-14	jimin	4 days ago	jimin
af6b189	경로수정	jimin	4 days ago	jimin



Github 고급 사용법2 - 복구하기

 Push Branch main

Push to branch in remote


Select a remote and the name the branch should have in the remote.

Source:

 main
  2fb7879 Revert "Rollback 시연용 Commit"

Destination:

Remote:

Branch:

☒ Configure upstream for push and pull

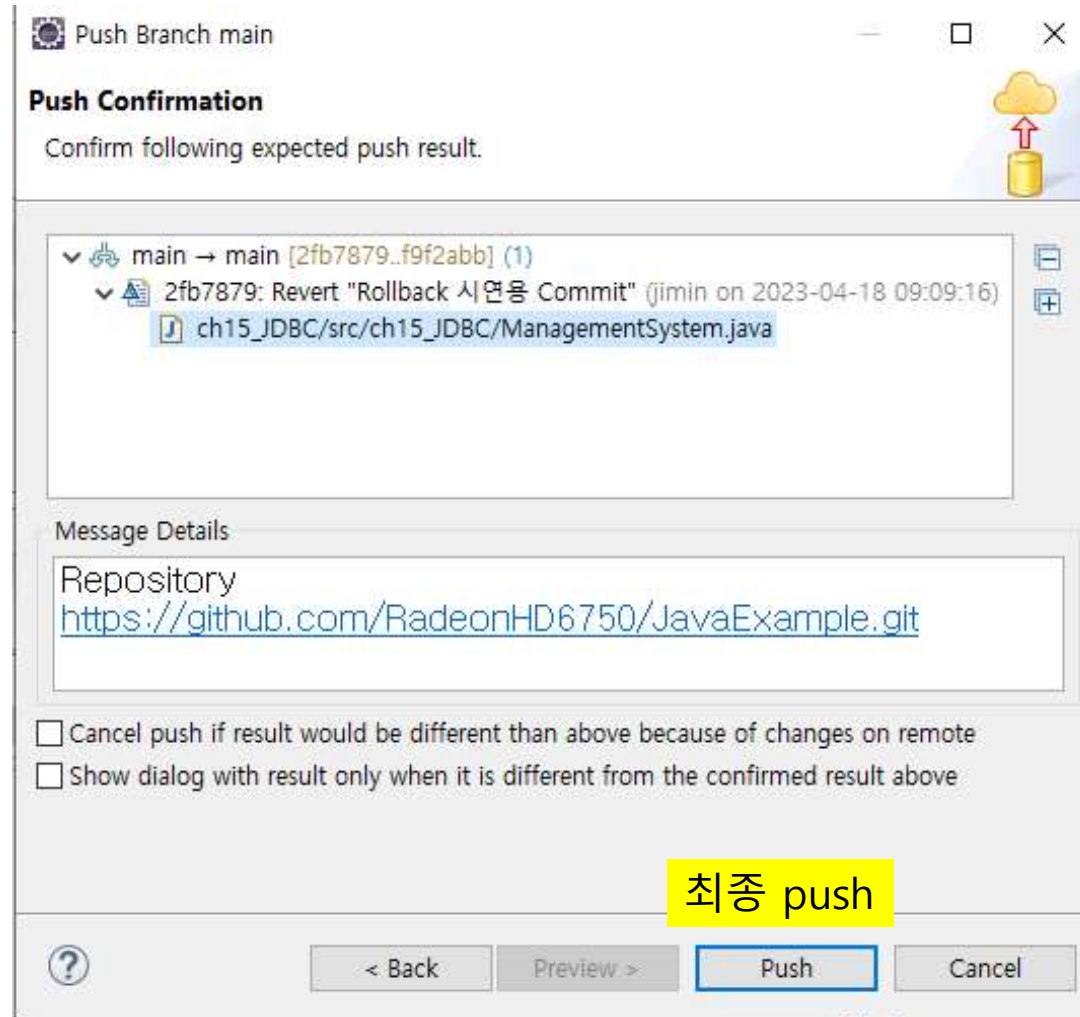
When pulling:

☐ Force overwrite branch in remote if it exists and has diverged

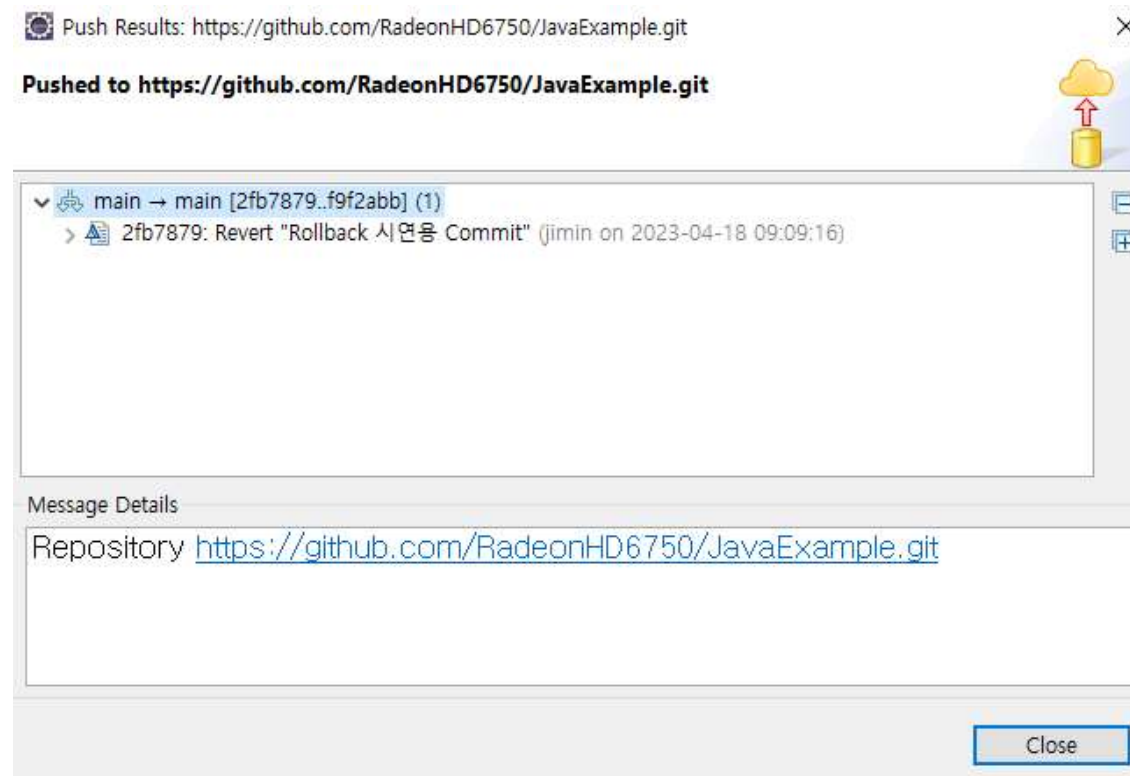
[Show advanced push dialog](#)

Preview

Github 고급 사용법2 - 복구하기



Github 고급 사용법2 - 복구하기



요렇게 뜨면 성공

Github 고급 사용법2 - 복구하기

```

import java.sql.*;

/*****
 *
 * 클래스명 | ManagementSystem
 *
 * 내용 | DB 관리 시스템 메인222
 *
 * 작성일 | 2023.04.12
 * 소속 | 개발1팀
 * 직급 | 주임
 * 성명 | 서지민
 *****/

public class ManagementSystem {

    public static void main(String[] args) {

        DBConnection dbConnection = new DBConnection();

        System.out.println("MariaDB 연결 테스트 = " + dbConnection.connectTestDB());

        DBManager dbManager = new DBManager();

        //01. 사용자 입력을 받아서
        String userId = "km29a1";
        String password = "jimin";

        //02. DB에 물어본다.
        HashMap<String, Object> loginResult = dbManager.loginTransaction(userId, password);

        System.out.printf("%s %s 로그인한 사람 %s \n\n", userId, password, loginResult.get("name"));

    }

}
    
```

원격 저장소를 확인해보면 반영되어 있다.

Github 고급 사용법3 – 최후의 보루

JAVA > JAVA Example

이름	수정한 날짜	유형	크기
JavaExample	2023-04-14 오후 4:38	파일 폴더	
jeong	2023-04-04 오후 2:05	파일 폴더	
jimin	2023-04-14 오전 10:08	파일 폴더	
RadeonHD6750.JAVAExample	2023-04-14 오전 10:06	파일 폴더	
ZIP JAVA Example-2023-04-14.zip	2023-04-14 오전 9:25	압축(ZIP) 파일	2,424,598...
ZIP JAVA Example-2023-04-17.zip	2023-04-17 오후 5:38	압축(ZIP) 파일	126,072KB
ZIP jimin.zip	2023-04-14 오전 9:46	압축(ZIP) 파일	20,681KB

복구시 Rollback이 모종의 원인으로 실패하였고, git 원격 저장소 마저도 사망했거나 잘 해야 혼수상태인 경우를 대비하는 방법이 있다.

때론 원시적인 방법이 확실할 때가 있다. 바로 프로젝트 폴더째로 버전별로 압축해 놓는 것이다. 권장하지 않는 회사도 있으나 결국 본인의 소스코드를 지킬 수 있는 것은 본인뿐이다.