



멋쟁이사자처럼



AT 남서울대

Git&GitHub

교육



Git&GitHub

CONTENTS

01

Git

02

Repository

03

명령어

04

Intellij Git 연동

05

VSCode Git 연동

06

Markdown

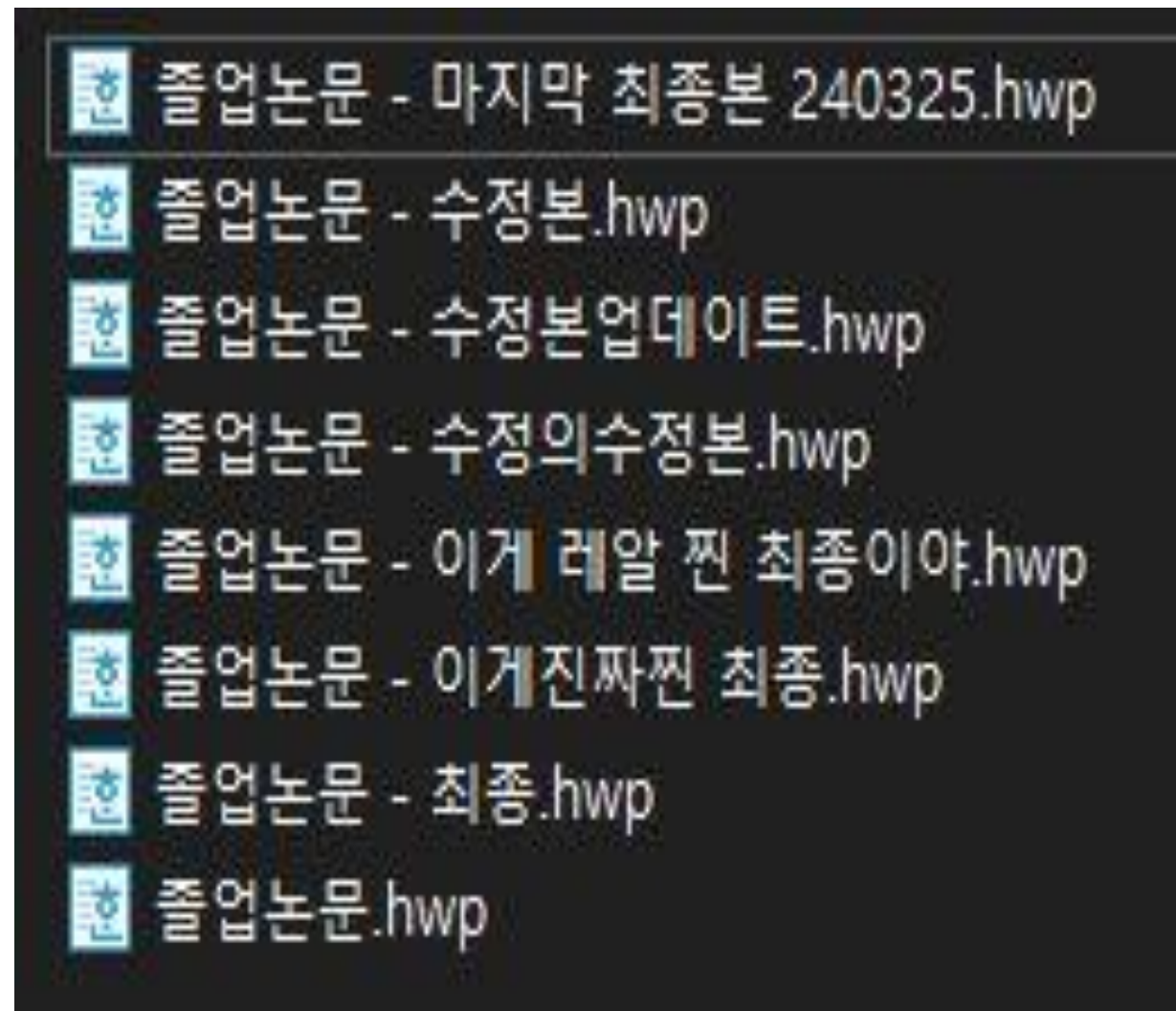


멋쟁이사자처럼
AT 남서울대



01 . Git

1.1 Git 필요성



문서 작업하다가 최종본이 무엇인지 모르겠어요..



버전 관리란?



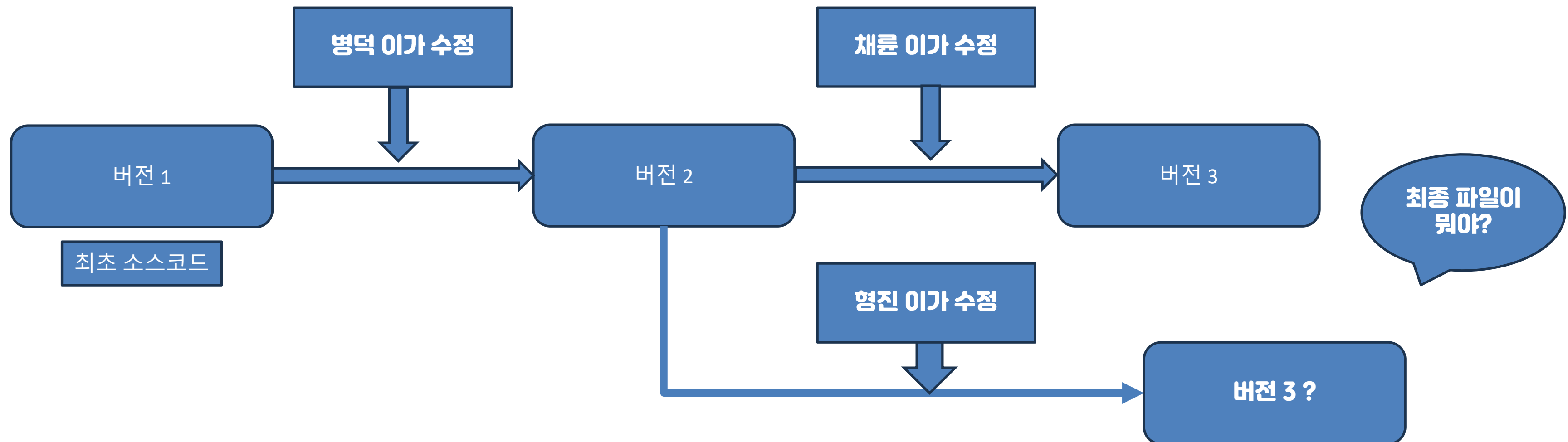
버전 관리

```
public class PrintTest2 {  
    public static void main(String[] args) {  
        System.out.print("여러분들");  
        System.out.println("반가워여");  
        System.out.println("깃 강의중입니다.");  
    }  
}
```



[Ctrl+Z]

버전 관리



팀 프로젝트에 참여하는 인원이 많을수록, 또 프로젝트 기간이 길수록 어느 파일이 최종 업데이트 파일인지 확인할 길이 막막해집니다. 그래서 여럿이 함께 작업하는 협업 프로젝트에서는 버전 관리를 해줘야 해요.

Git

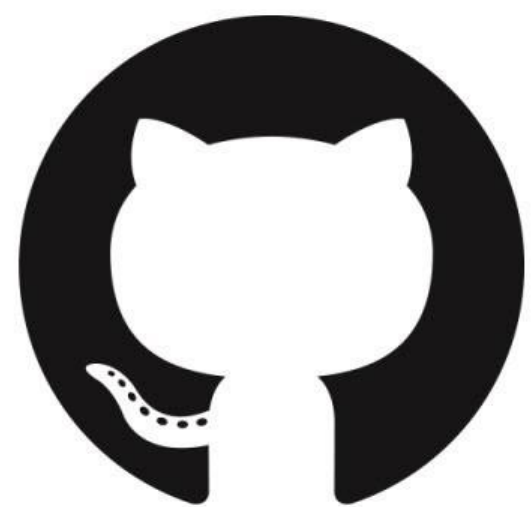
Git 이란?

- 코드 저장소
 - 이력을 관리하는 저장소
 - Version Control System
 - Code를 저장 및 저장지점으로 되돌아갈 수 있도록 해주는 시스템
 - 분산 버전 관리 시스템
- => 분할 작업자들과 저장소에 분산해서 저장 관리





GitHub



GitHub

GitHub 란?

- 코드저장소(git) 을 웹에 옮겨 놓은 원격저장소

장점?

- 프로젝트 소스코드 공유 가능
- 코드 복구 가능



분산버전관리

버전 관리란 ?

특정 시점의 버전을 가져올 수 있다.
파일을 시간에 따라 기록할 수 있다.

장점 ?

- 1. 소스코드 변경이력 확인 가능**
- 2. 이전에 저장한 버전과 비교 가능**
- 3. 이전 버전으로 되돌리기 가능**
- 4. 변경이력 충돌 위험이 없음**
- 5. 덮어쓰기 충돌 위험 없음**



멋쟁이사자처럼
AT 남서울대

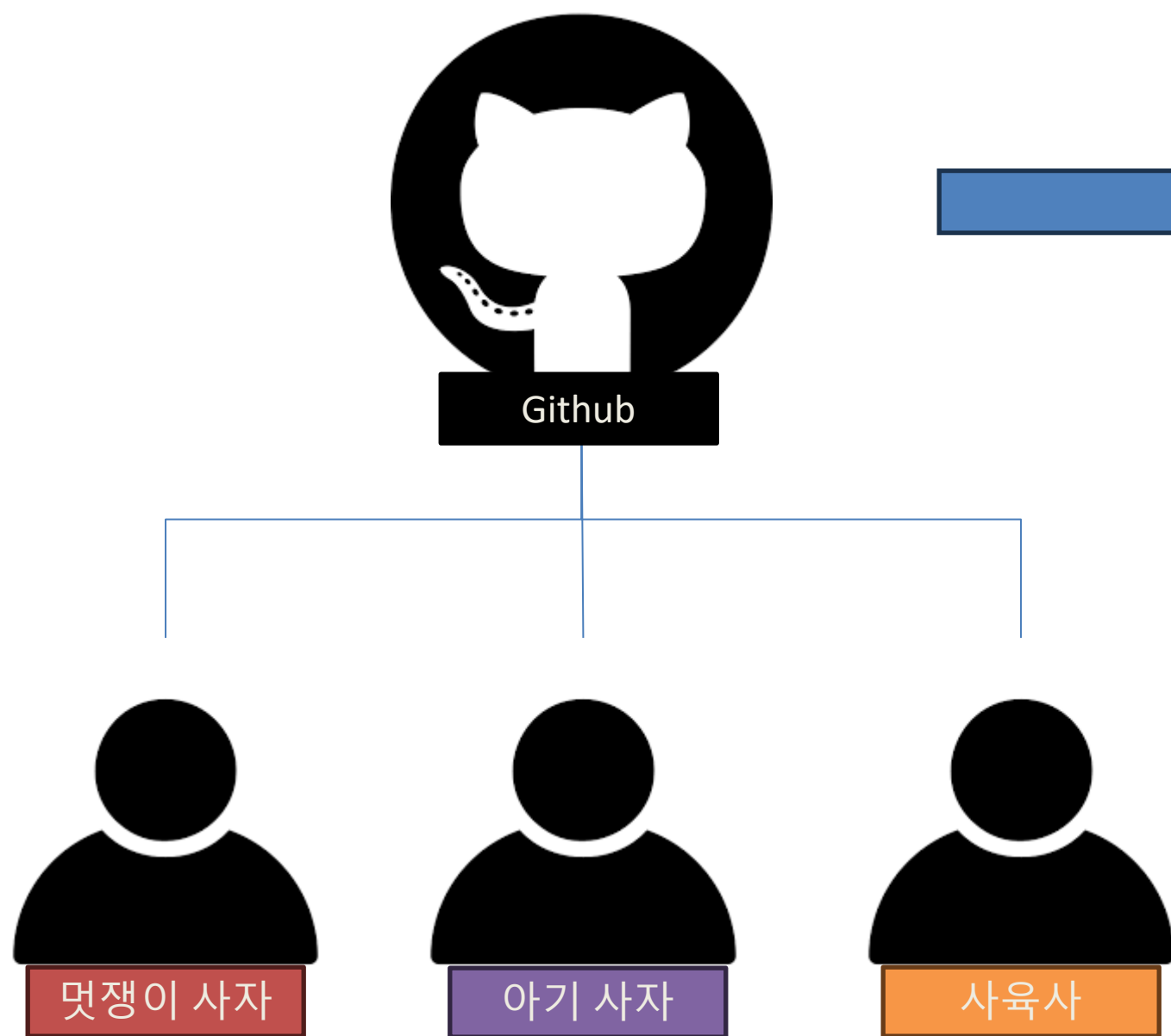


02 . Repository



저장소란 ?

Git 으로 관리하는 프로젝트 저장소이다.
다음과 같이 두 가지 종류로 나뉜다.



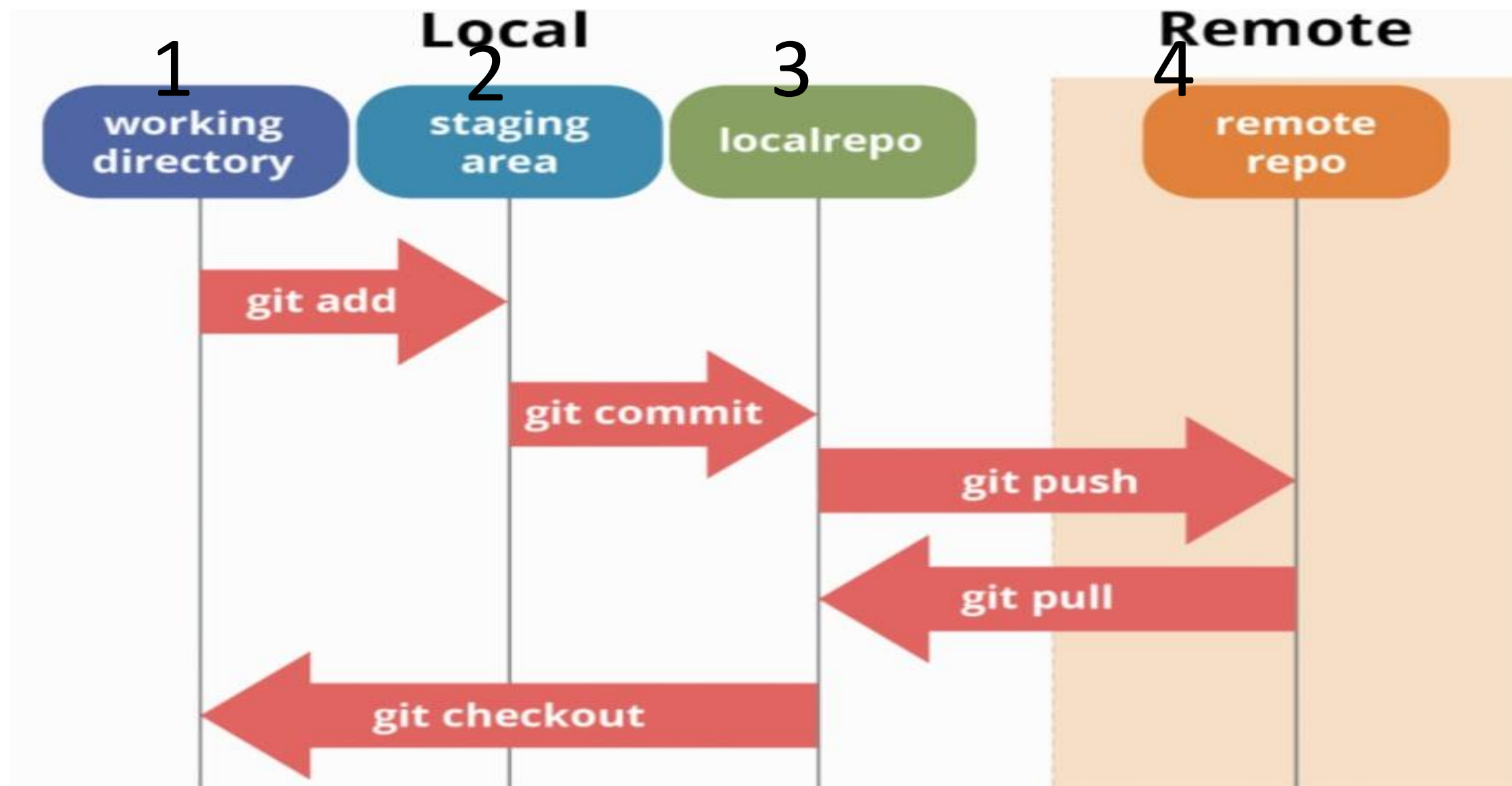
원격 저장소 (Remote Repository) ?

파일이 원격저장소 전용 서버에서 관리 받아
여러 사람들과 함께 공유할 수 있는 저장소

로컬 저장소 (Local Repository) ?

내 컴퓨터에서 만 파일을 저장할 수 있는
개인 저장소

저장소란?



1. Working directory

로컬 작업 디렉토리 (내 자신 컴퓨터)

2. Staging area

원격 저장소 가기 전에 임시로 보관할수 있는 파일장소

3. Localrepo

로컬에 저장된 파일을 push할 경우 원격저장소로 반영
터미널 에서 명령어 를 입력하면 전송 하는 역할

4. Remoterepo

원격 저장소 (gutHub)

Git

Git 설치

설치 경로

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

설치 방법

<https://xangmin.tistory.com/102>

기본 설정

git config --global user.name "이름"

git config --global user.email "깃허브 이메일"

설정 확인

git config --list

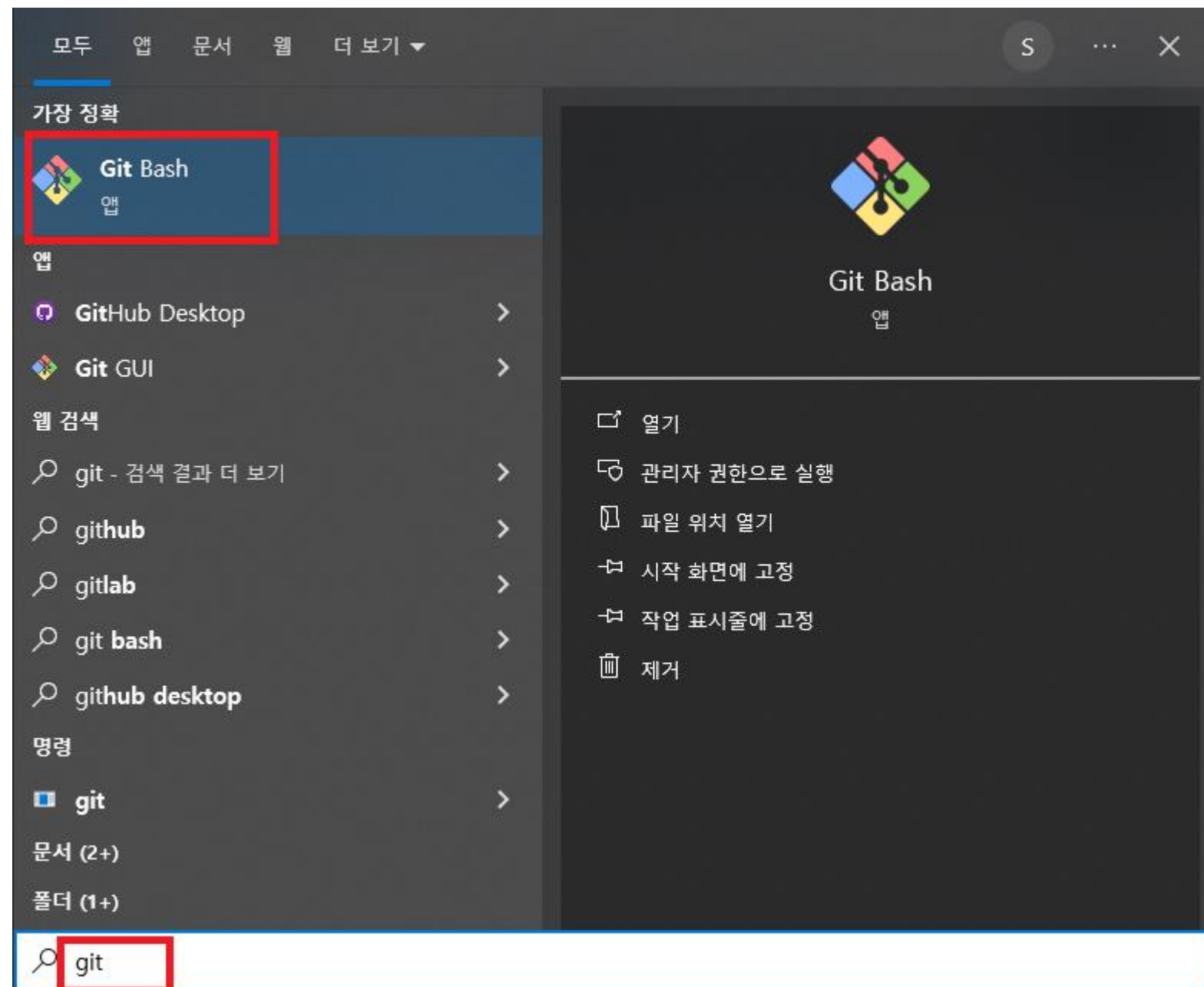
```
@DUDU MINGW64 ~ (master)
$ git config --global user.name "Jeongbubu"

@DUDU MINGW64 ~ (master)
$ git config --global user.email "zhfl1234@gmail.com"

@DUDU MINGW64 ~ (master)
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Jeongbubu
user.email=zhfl1234@gmail.com
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
remote.origin.url=https://github.com/Like-Lion-NSU-24/DeveloperTest-BE.git
...skipping...
```

Git

Git Bash 사용방법



올리고자 하는 폴더에서 마우스 우클릭 후 Git Bash Here 클릭한다.

혹은 윈도우 키 (Windows key) 검색 창에서 git 검색 해서 실행한다.

Git

Git Bash 사용방법

```

C:\> cd C:\Program Files\Git\mingw64\bin
C:\Program Files\Git\mingw64\bin> git config --global user.name "Jeongbubu"

C:\Program Files\Git\mingw64\bin> git config --global user.email "zhfl1234@gmail.com"

C:\Program Files\Git\mingw64\bin> git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Jeongbubu
user.email=zhfl1234@gmail.com
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
remote.origin.url=https://github.com/Like-Lion-NSU-24/DeveloperTest-BE.git
...skipping...

```

기본 설정

git config --global user.name “이름”

git config --global user.email “깃허브 이메일”

설정 확인

git config --list



GitHub

JeongBuBu

Overview

Repositories2

Projects

Packages

Stars

QType to search

>_

+

정두두

JeongBuBu

Edit profile

Organizations

Popular repositories

dudu

Public

멋쟁이 사자처럼

30 contributions in the last year

Contribution settings

2024

2023

2022

Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar

Mon

Wed

Fri

Learn how we count contributions

Less

More

Contribution activity

March 2024

Created 4 commits in 1 repository

Like-Lion-NSU-24/DeveloperTest-BE4 commits

Show more activity

Seeing something unexpected? Take a look at the GitHub profile guide.

© 2024 GitHub, Inc.

Terms

Privacy

Security

Status

Docs

Contact

Manage cookies

Do not share my personal information



GitHub

The screenshot shows the GitHub 'New' page for a user named JeongBuBu. The page is dark-themed. At the top, there's a navigation bar with the GitHub logo, the user's name, and a search bar. Below the navigation bar, there's a row of tabs: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The 'Code' tab is selected. The main content area is titled 'dudu' and 'Public'. It features several cards: 'Set up GitHub Copilot', 'Add collaborators to this repository', and 'Quick setup — if you've done this kind of thing before'. The 'Quick setup' card includes a code block with the following commands:

```
echo "# dudu" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/JeongBuBu/dudu.git
git push -u origin main
```

 Below this, there's another code block for pushing an existing repository:

```
git remote add origin https://github.com/JeongBuBu/dudu.git
git branch -M main
git push -u origin main
```

 At the bottom, there's a section for importing code from another repository. On the right side, a dropdown menu is open, showing options: 'New repository', 'Import repository', 'New codespace', 'New gist', and 'New organization'. The 'New repository' option is highlighted with a red box.

GitHub

Repository name – READEM 작성

Private 클릭

Create a new repository

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

Required fields are marked with an asterisk ().*

Owner *

JeongBuBu

Repository name *

READEM

✓ READEM is available.

Great repository names are short and memorable. Need inspiration? How about **fantastic-octo-doodle** ?

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:

☐ Add a README file

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

Add .gitignore

.gitignore template: None

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

Choose a license

License: None


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

ⓘ You are creating a private repository in your personal account.


Create repository




GitHub

 **READEM** Private

Unwatch 1 Fork 0 Star 0

**Set up GitHub Copilot**
Use GitHub's AI pair programmer to autocomplete suggestions as you code.
[Get started with GitHub Copilot](#)

**Add collaborators to this repository**
Search for people using their GitHub username or email address.
[Invite collaborators](#)

Quick setup — if you've done this kind of thing before

[Set up in Desktop](#) or [HTTPS](#) [SSH](#) [Copy](#)

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# README" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/JeongBuBu/READEM.git
git push -u origin main
```

[Copy](#)

...or push an existing repository from the command line

```
git remote add origin https://github.com/JeongBuBu/READEM.git
git branch -M main
git push -u origin main
```

[Copy](#)

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.



멋쟁이사자처럼
AT 남서울대



03 . 명령어



3. Add

Add 명령어 ? 저장소 에 파일 추가 (Staged)

[Git bash 이용 방법]

방법 1 – 디렉토리 안 모든 파일 추가

: - git add .

방법 2 – 특정 파일만 추가

: - add [경로/파일 이름]

3. Commit

Commit 명령어 ?

Staging 한 파일 커밋.

커밋 메시지는 해당 작업에서 추가/변경 된 사항으로 간략하게 작성 한다.
타인이 알아볼 수 있도록 작성 하도록 하고 영어로도 작성 가능하다.

[Git bash 이용 방법]

방법 – **git commit -m ‘ 파일 에 대한 커밋 메시지 ‘**

3. Remote

Remote 명령어 ?

Git remote 는 로컬 저장소 와 원격 저장소 간의 연결을 의미합니다.

[Git bash 이용 방법]

방법 – **git remote add [원격 저장소의 이름을 설정하는 부분] [원격저장소 URL]**

* [별칭]은 원격 저장소의 이름을 설정하는 부분입니다. 일반적으로 **origin**이라는 이름을 많이 사용합니다.

* [원격 저장소 URL]는 **https://github.com/사용자이름/저장소이름.git** 형식으로 구성됩니다.

예시) –

git remote add origin https://github.com/사용자이름/저장소이름.git



3. Push

Push 명령어 ?

원격 저장소로 이동하는 명령어 Push

[Git bash 이용 방법]

방법 – **git push [원격 저장소 이름] [원격 저장소 브랜치 이름]**

예시) **git push origin master**

=> 로컬 저장소의 변경사항을 원격 저장소(origin)의 브랜치(master)에 반영 하겠다는 명령어

3. pull

pull 명령어 ?

Git pull 은 다른 사람이 PR을 통해서 코드를 업데이트 했거나, 아니면 github 를 통해서 Commit 했을때 그 내용을 클라이언트로 내려받는 명령어 입니다.

[Git bash 이용 방법]

방법 - `git pull [원격 저장소 이름] [원격 저장소 브랜치 이름]`
예시) `git pull origin master`

origin은 원격 저장소 이름이고, master은 현재 사용하는 컴퓨터의 브랜치 이름입니다.



주의사항

git pull을 할 때는 깃허브의 유저이름과 비밀번호를 쳐야하는 경우가 많습니다. 매번 비밀번호를 치기 귀찮다면 `git config --global credential.helper 'store --file 경로'` 하면 됩니다. 해당 경로에 비밀번호가 저장된 파일이 생성됩니다. 단, **파일로 저장되는만큼 보안에 취약하기 때문에 주의해야 합니다.**

3. Fetch

Fetch 명령어 ?

패치(Fetch) 는 원격 저장소에 있는 프로젝트를 로컬 저장소에 가져오는 것입니다.
풀(Pull)과는 다르게 로컬 저장소에 있는 프로젝트를 병합하지 않고 가져오기만 합니다.
원격 저장소에 있는 변경사항을 확인해 볼 경우 패치를 사용합니다.

[Git bash 이용 방법]

방법 - `git fetch <원격 저장소 이름>`
예시) `git fetch origin`

3. Branch

Git 에서 Branch (브랜치) 란 단어 그대로 의 의미 처럼 독립적으로 어떤 작업을 진행하기 위한 개념입니다. 개발을 하다 보면, 한 페이지 안에 여러 기능을 따로 구현하거나, 이전 코드와 비교를 위해 여러 코드를 복사해야 하는 일이 자주 있습니다. Git의 브랜치를 활용하면, 코드를 통째로 복사한 후, 원래 코드에 영향을 주지않고 독립적으로 개발할 수 있습니다.

[Git bash 이용 방법]

git branch

브랜치를 생성하는 명령어는 다음과 같습니다.

`git branch { 브랜치명 }`

새로운 브랜치를 생성하면, 기반이 되는 브랜치(부모 브랜치)의 버전을 그대로 복사합니다. 커밋 내역(log)도 부모 브랜치와 같게 됩니다.

브랜치를 이동하는 명령어는 다음과 같습니다.

`git checkout { 브랜치명 }`

두 명령어를 합친, 즉 브랜치를 생성함과 동시에 이동하는 명령어는 다음과 같습니다.

`git checkout -b { 브랜치명 }`

수많은 명령어가 존재 합니다. 이것이 궁금하면, 노션 페이지 에서 참고 하세요.

3. checkout

checkout 명령어 ?

Git에서 현재 작업 중인 브랜치를 변경하거나 파일의 변경 사항을 취소하여 이전 상태로 되돌리는 명령어입니다. 주로 브랜치 전환 및 파일 복원에 사용됩니다.

[Git bash 이용 방법]

방법 – `git checkout <브랜치 이름>`
예시) `git checkout origin`

수많은 명령어가 존재 합니다. 이것이 궁금하면, 노션 페이지 에서 참고 하세요.



멋쟁이사자처럼
AT 남서울대

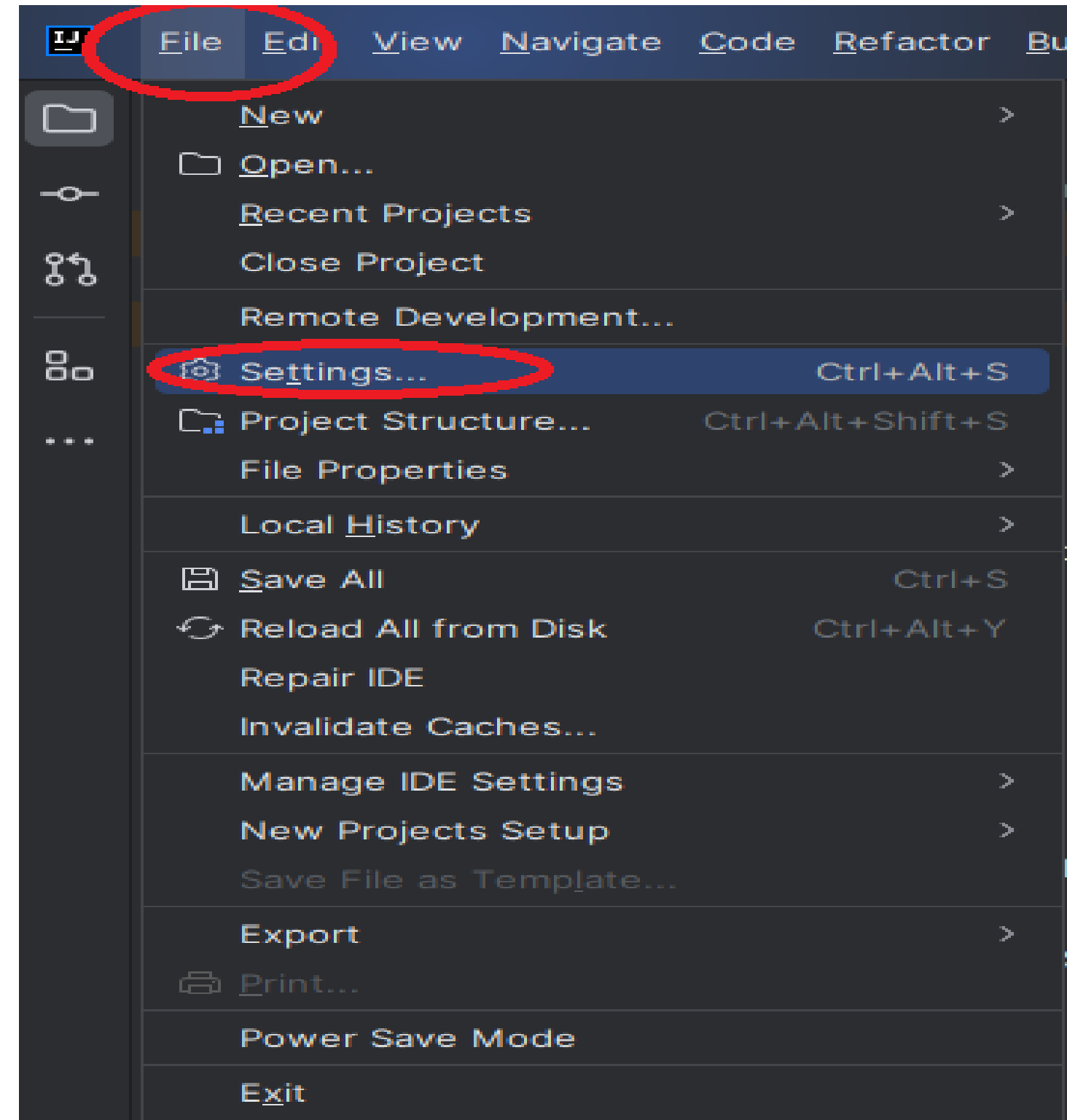


04. 실습 및 연동



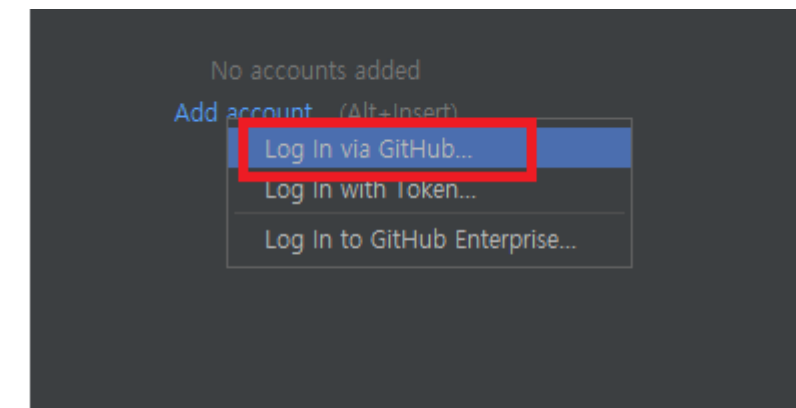
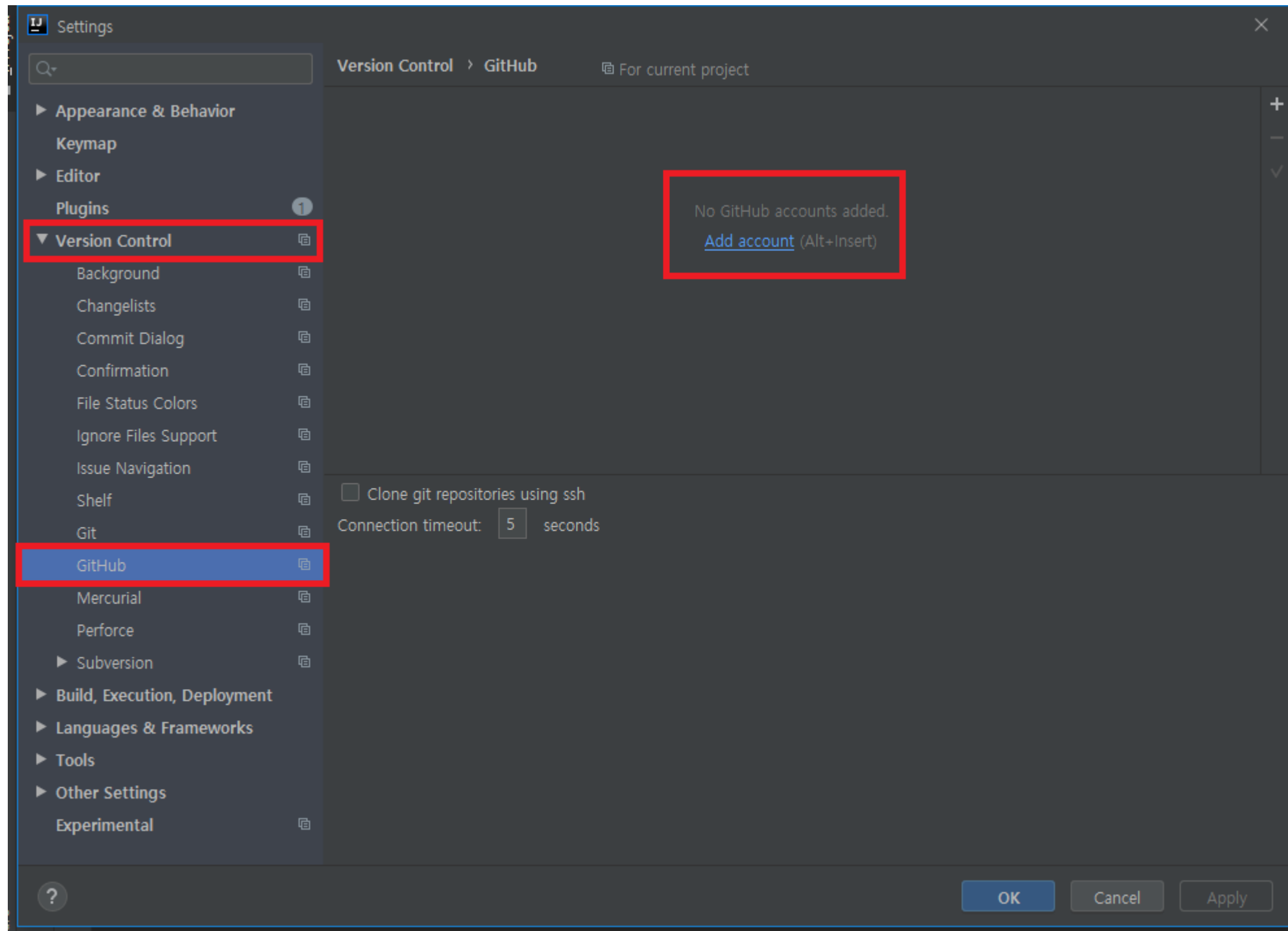
GitHub

IntelliJ GitHub 연동



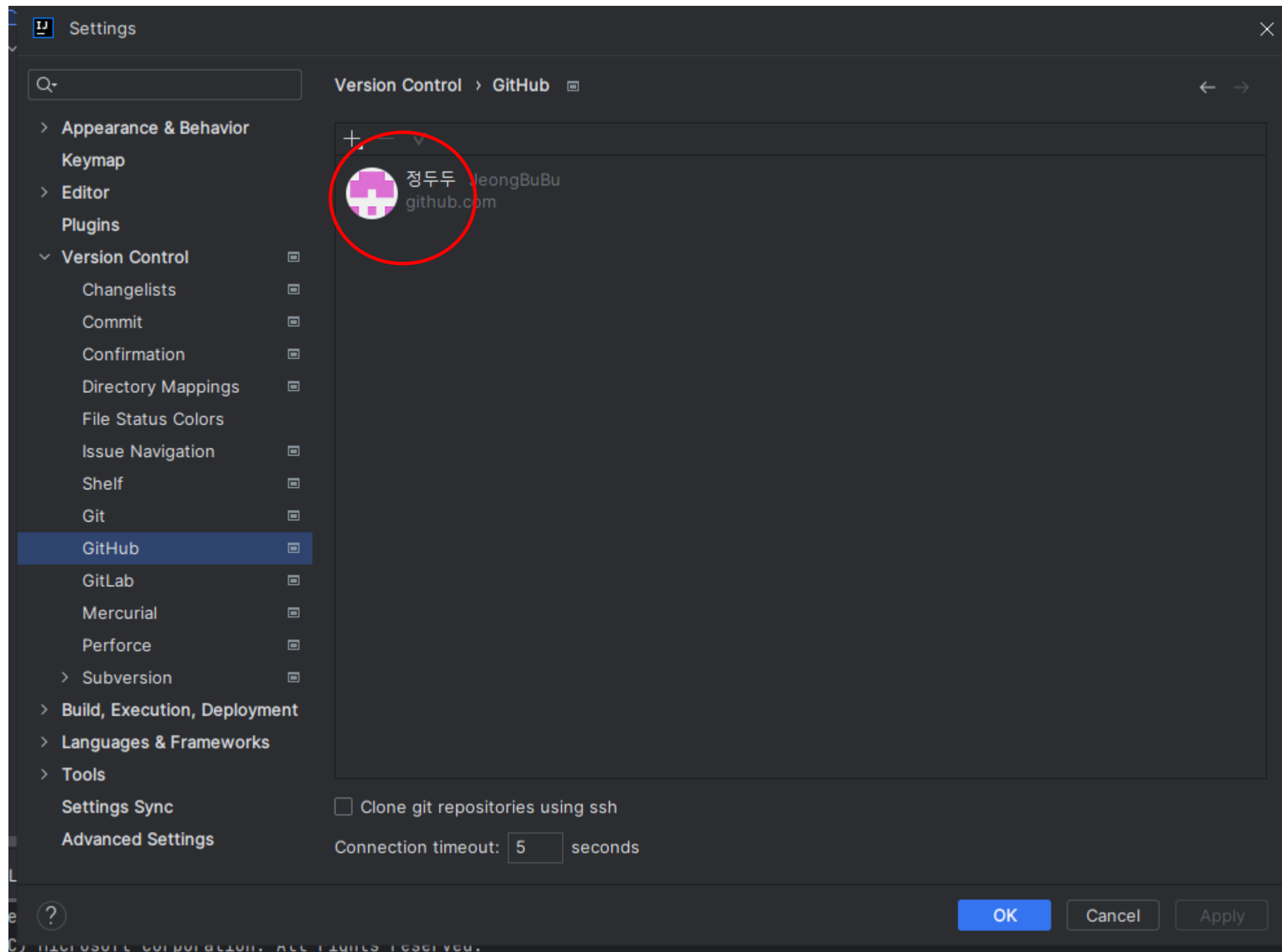
GitHub

IntelliJ GitHub 연동



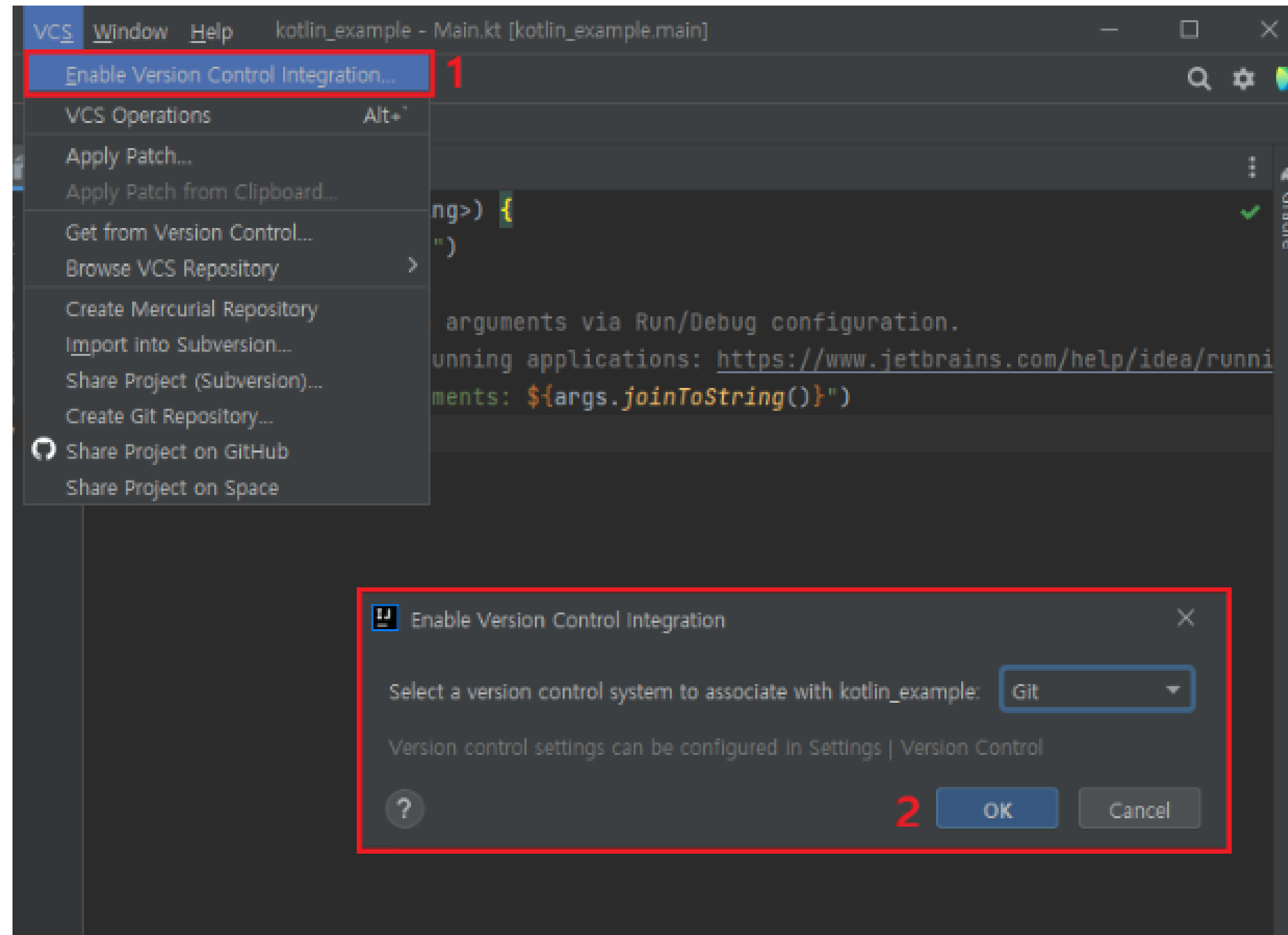
GitHub

IntelliJ GitHub 연동



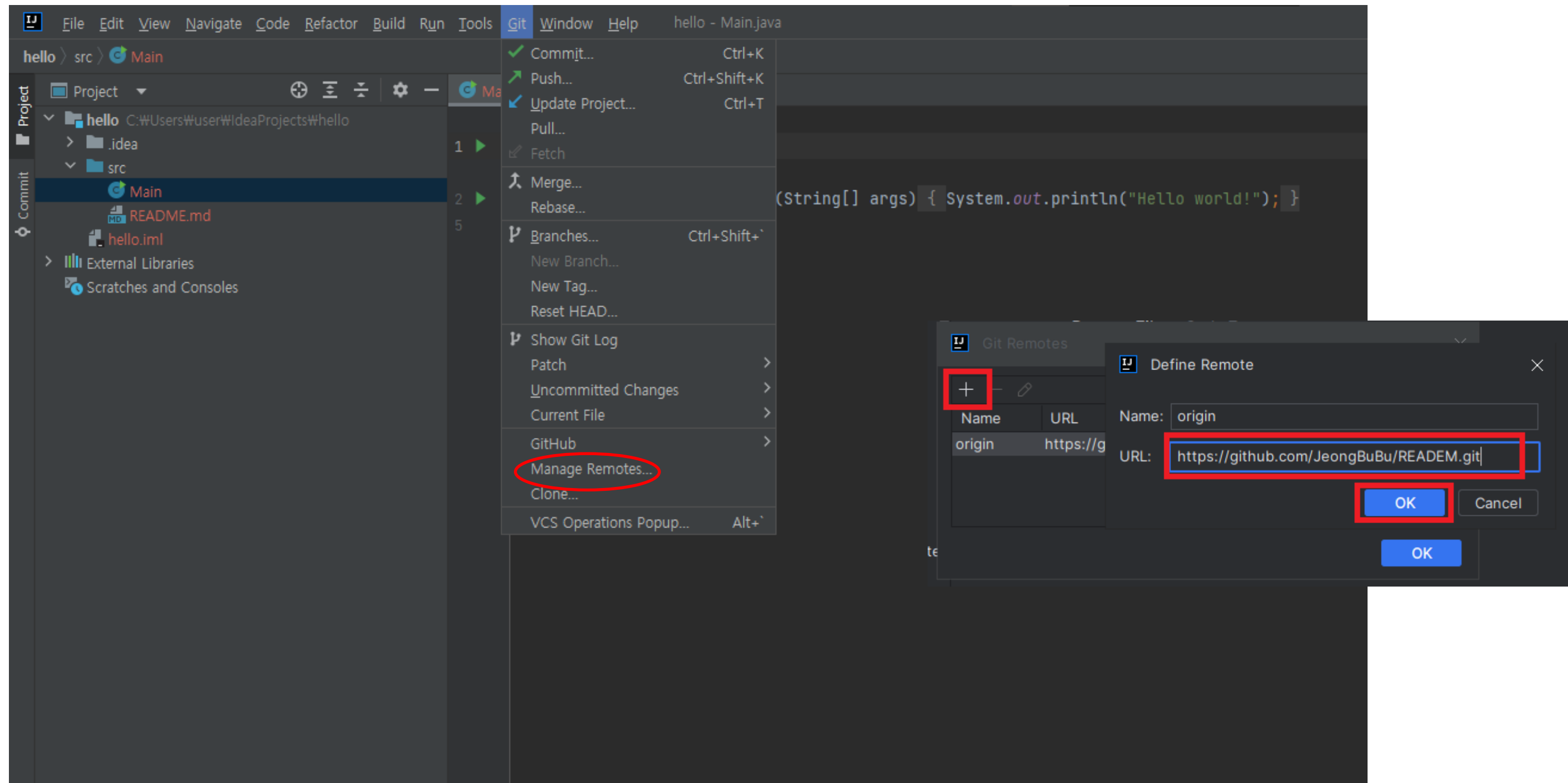
GitHub

IntelliJ GitHub 연동



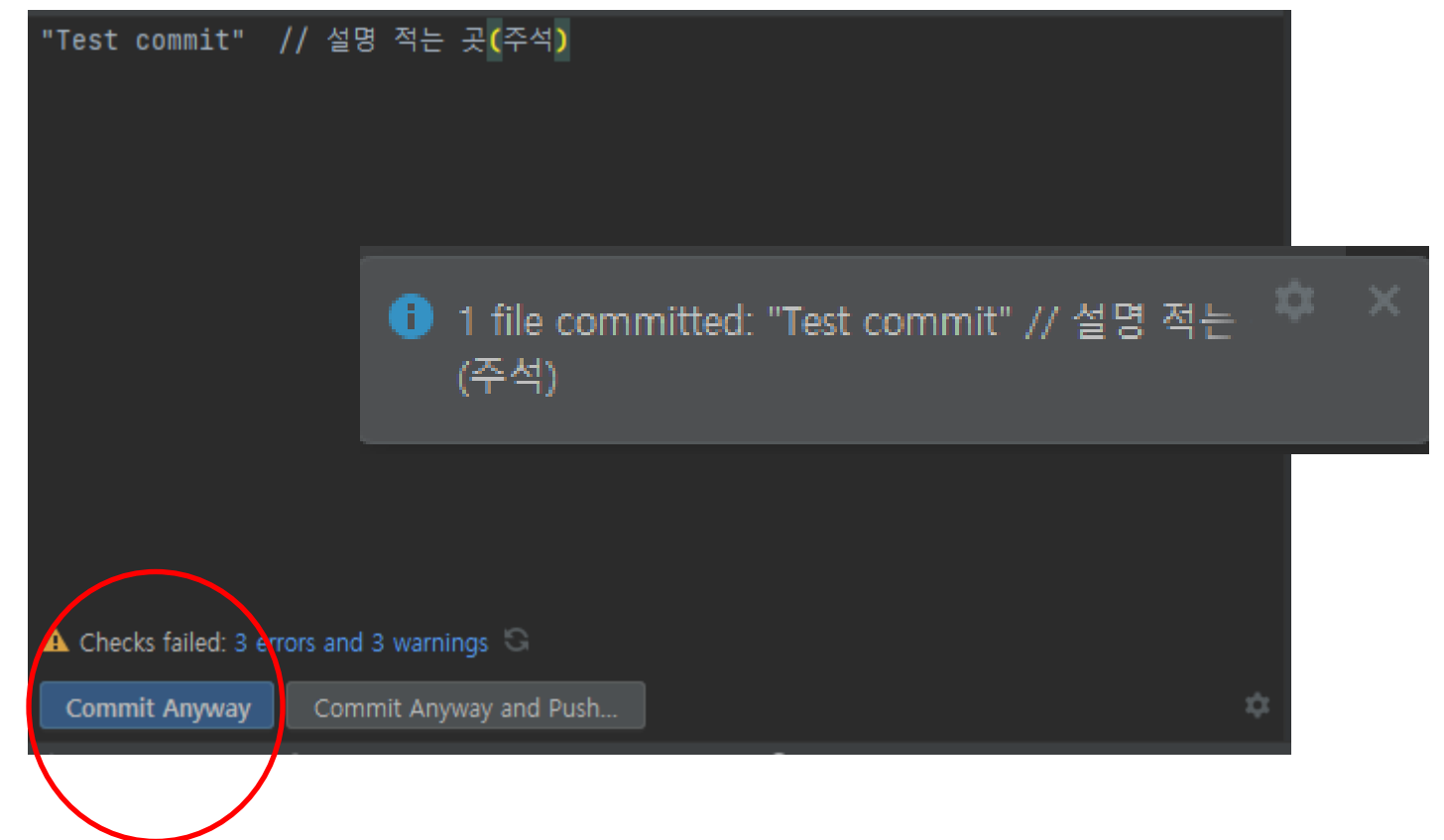
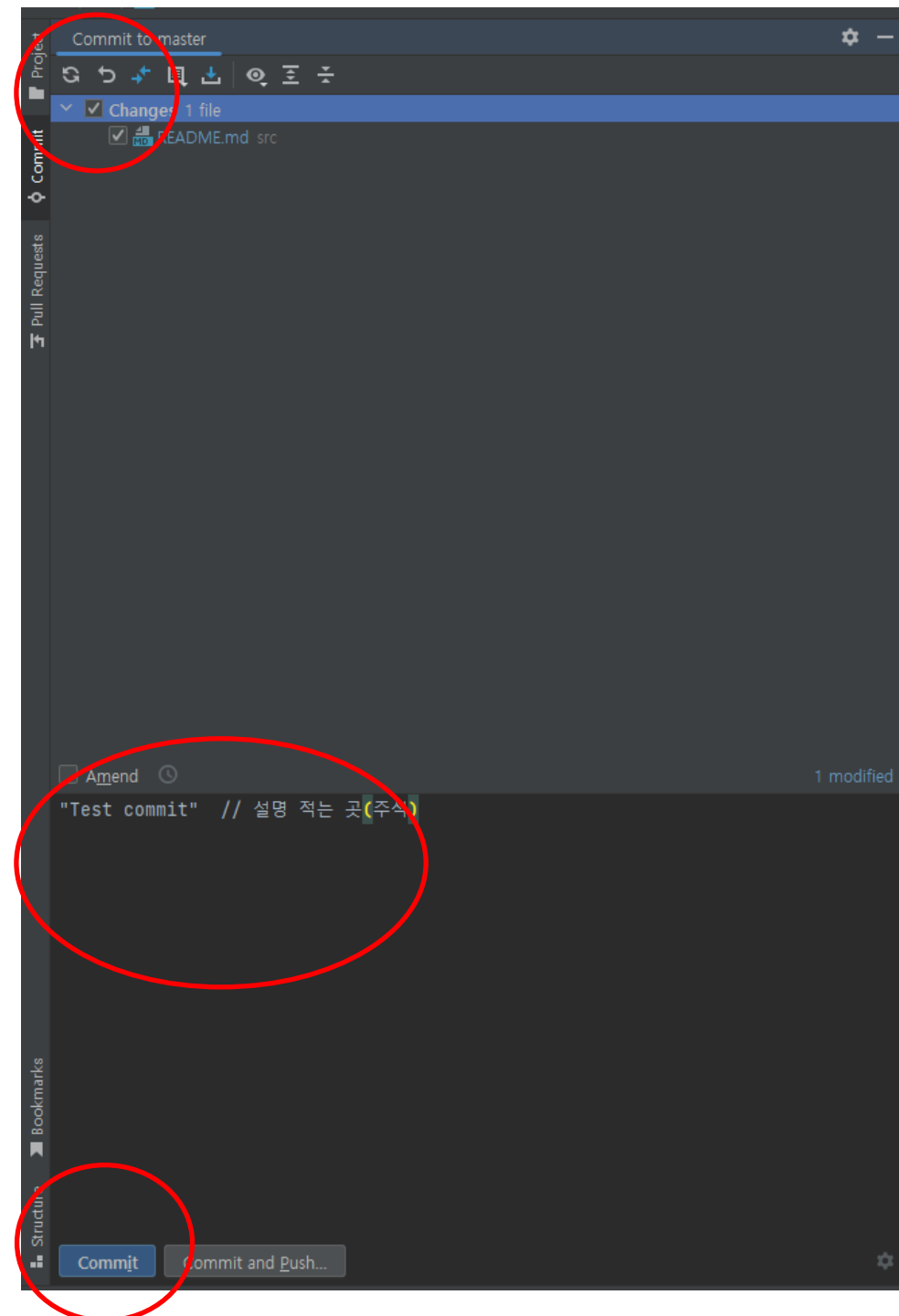
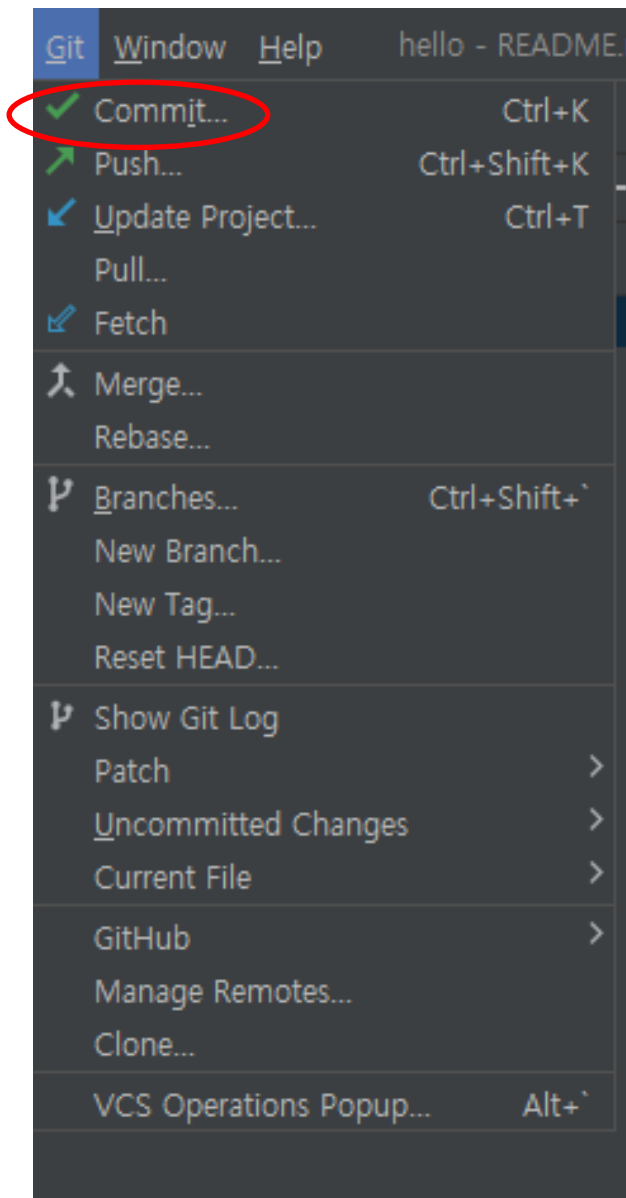
GitHub

IntelliJ GitHub 연동



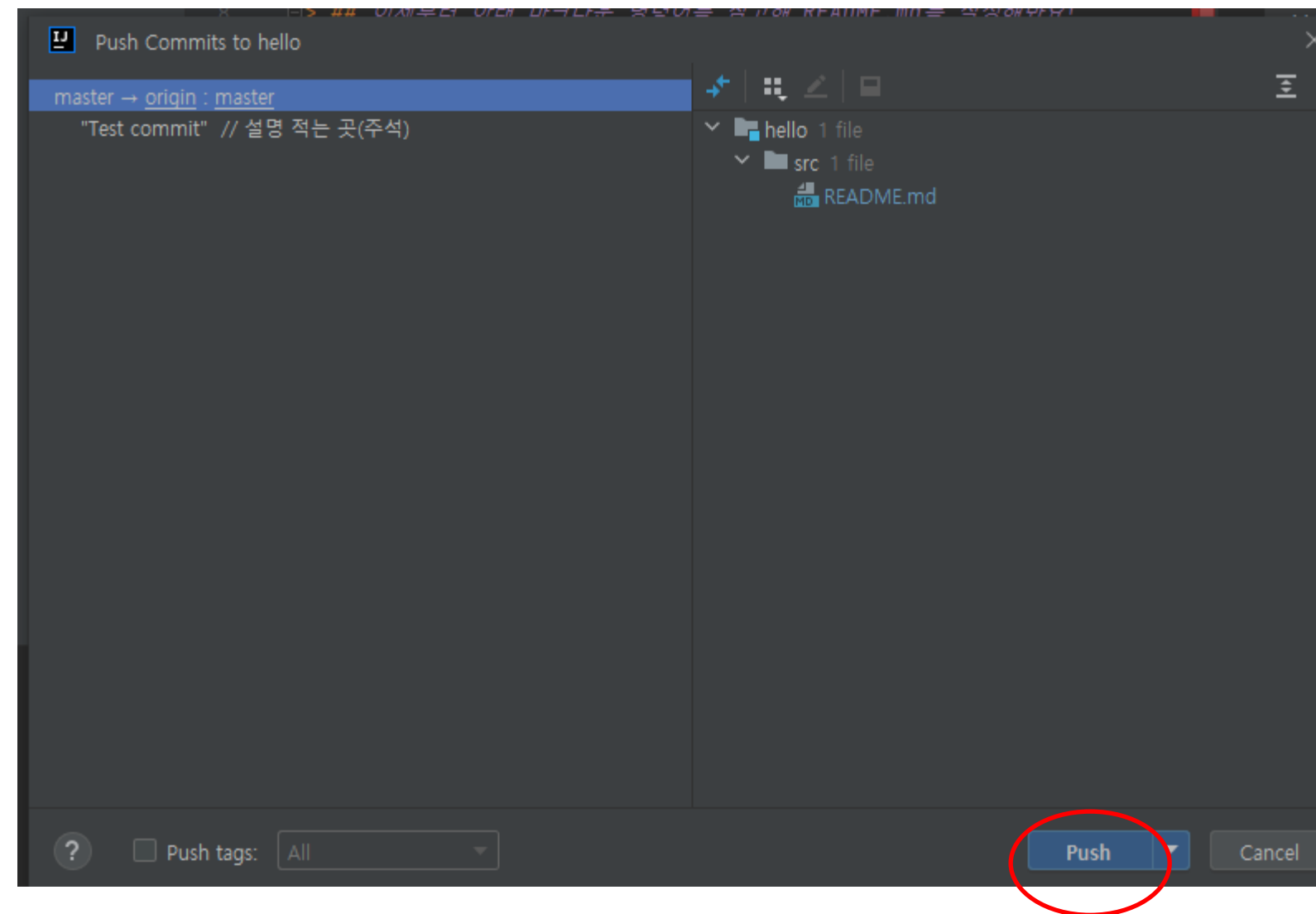
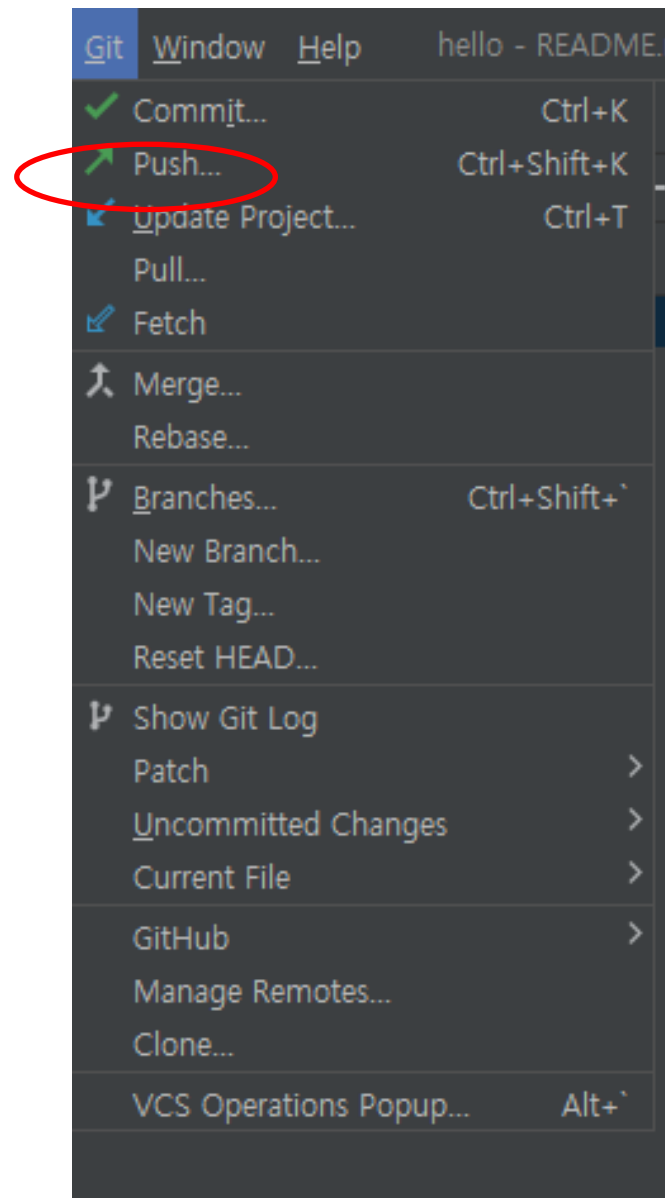
GitHub

IntelliJ GitHub 연동



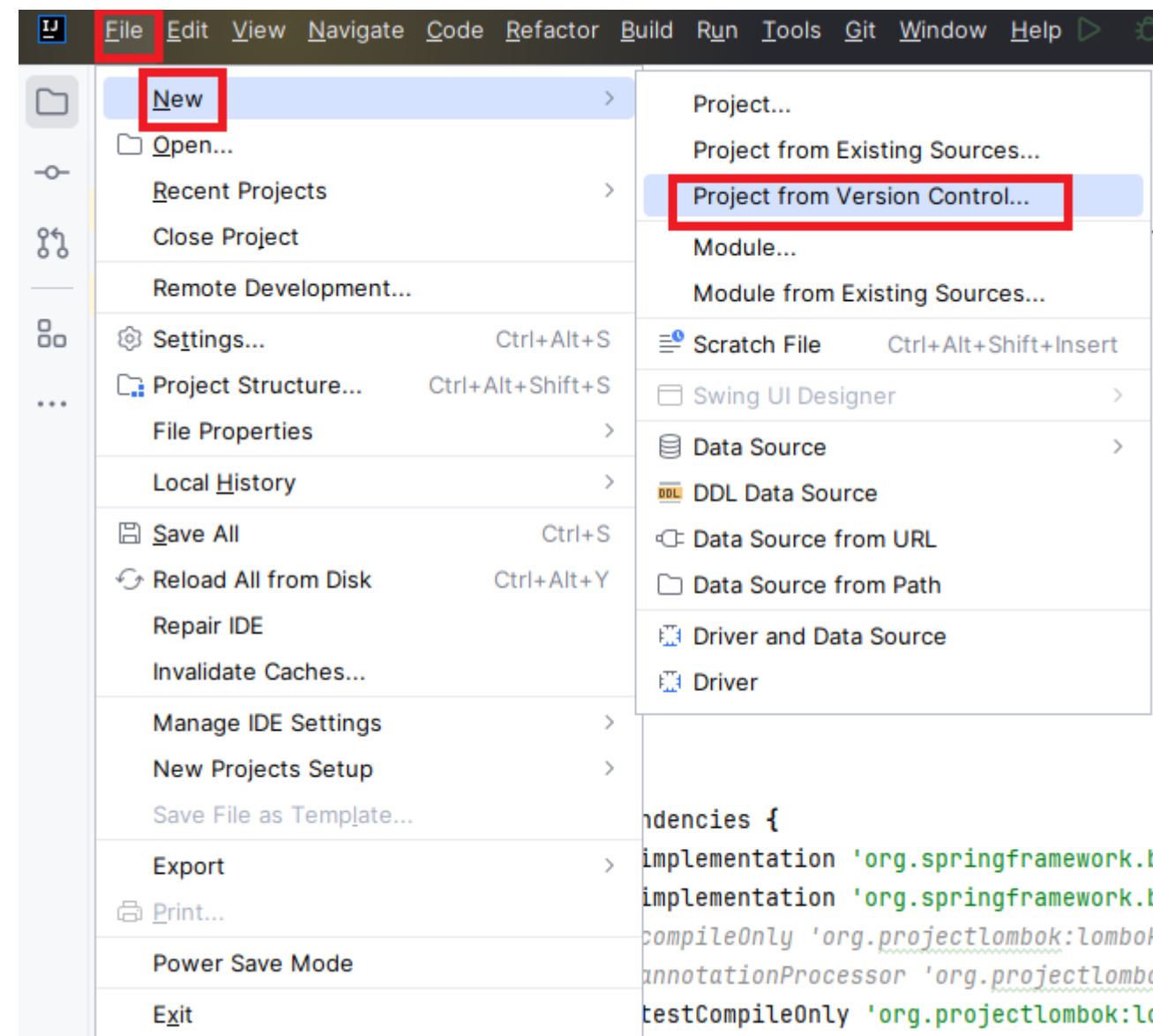
GitHub

IntelliJ GitHub 연동





Intellij Git clone 하기

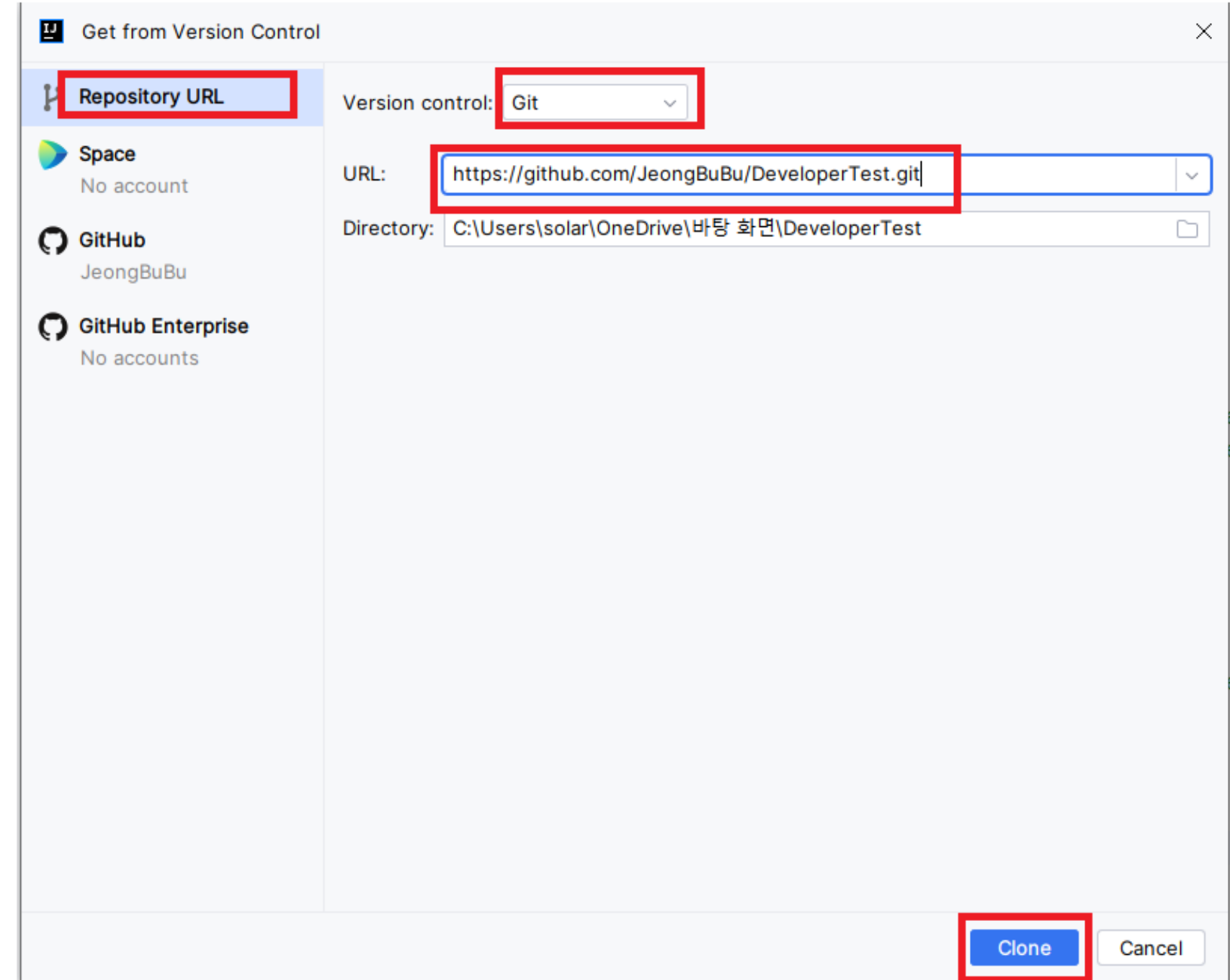


GitHub

IntelliJ Git clone 하기

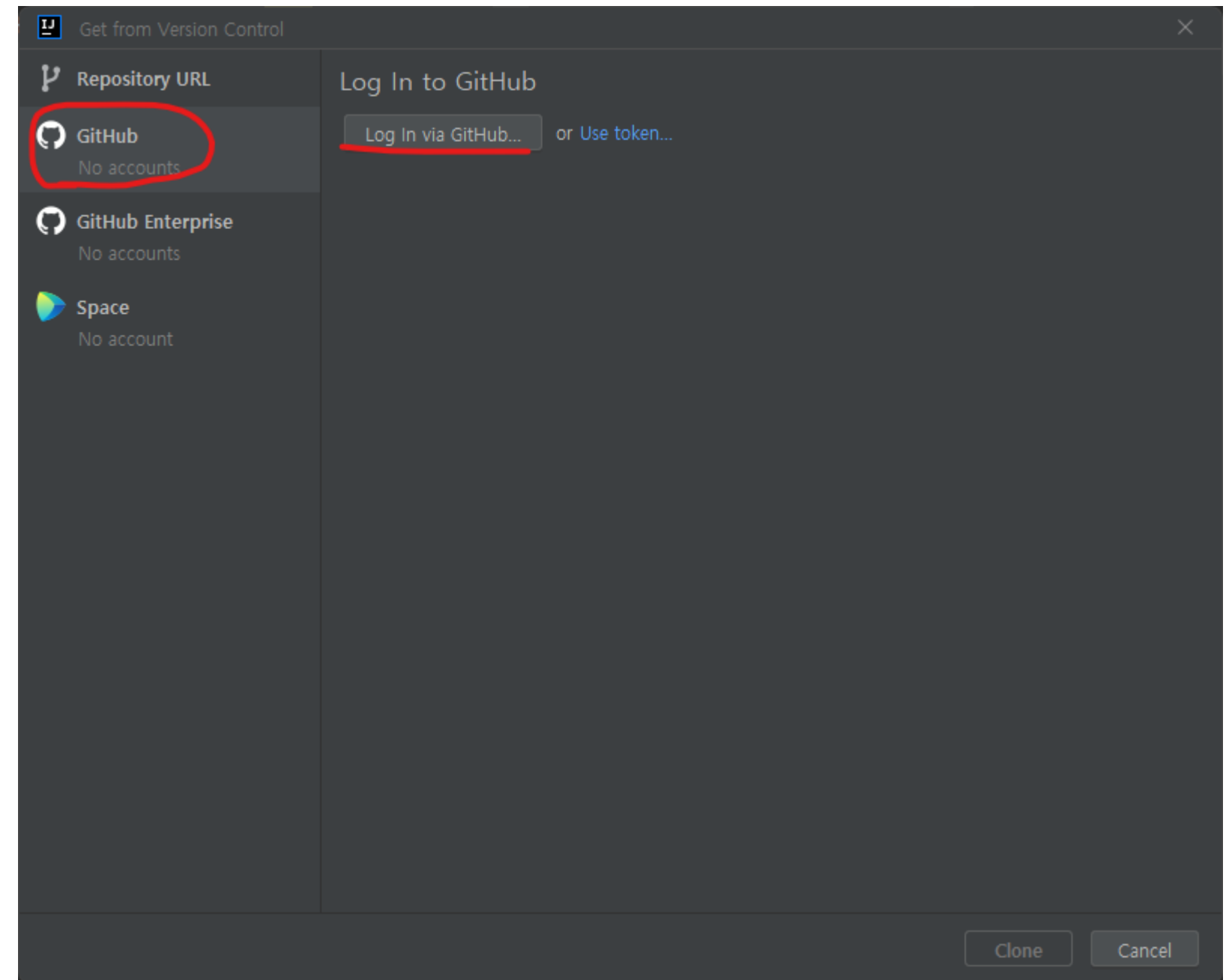
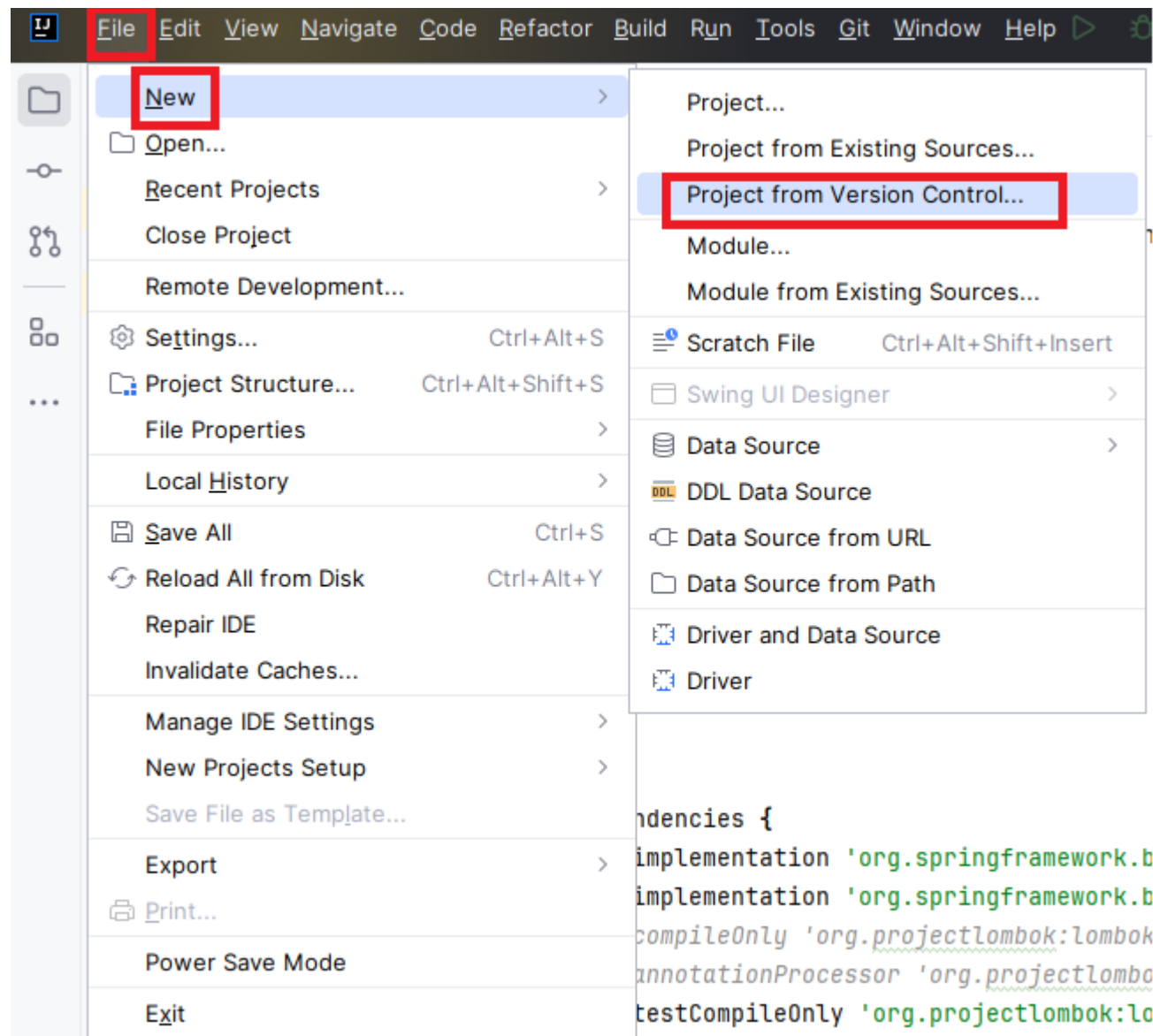
Repository URL 탭

- **Version control:** Git
- **URL:** 복사한 Git 주소를 붙여넣기 합니다.
- **Directory:** Git 소스를 Clone할 디렉토리를 지정합니다.



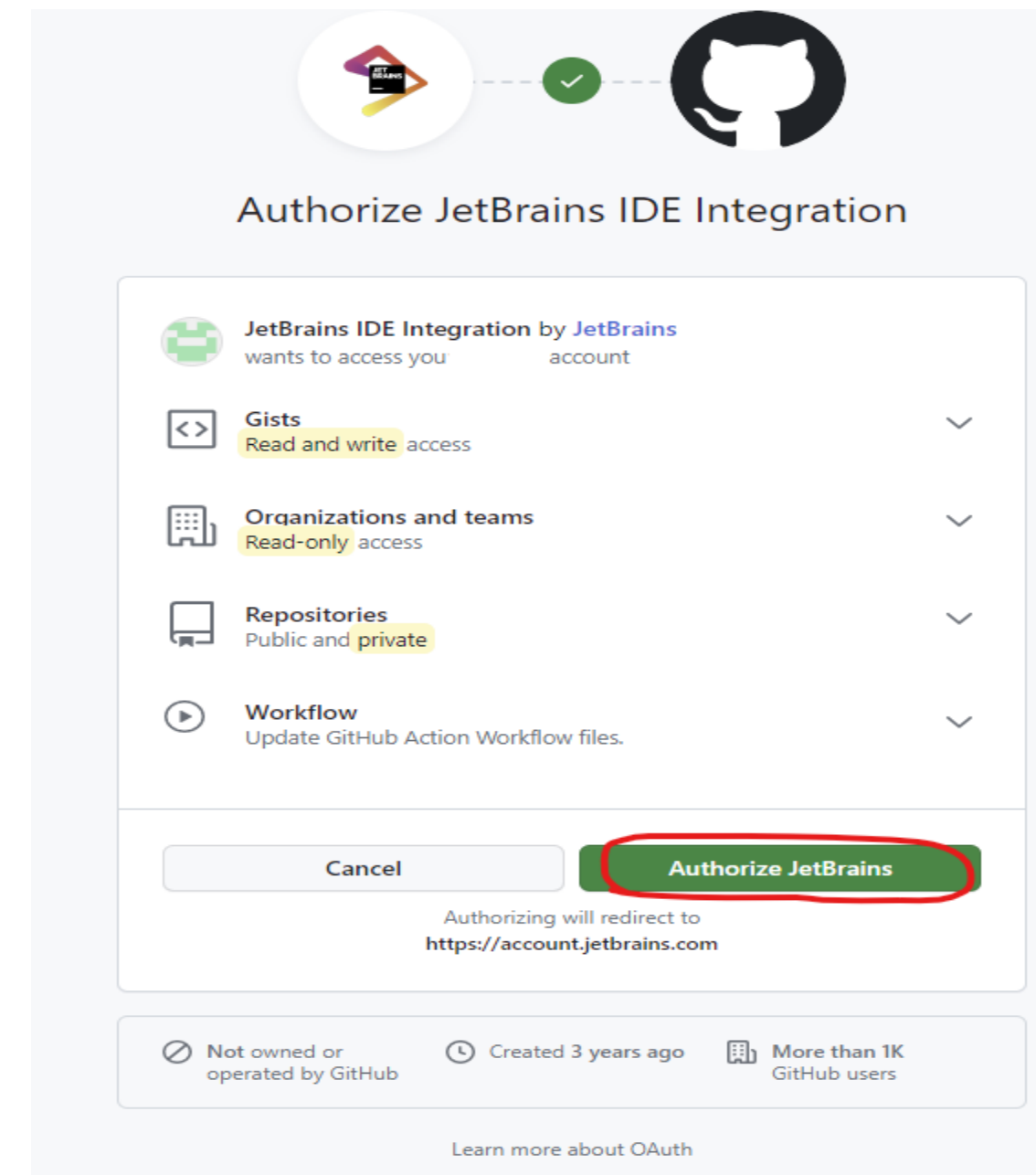
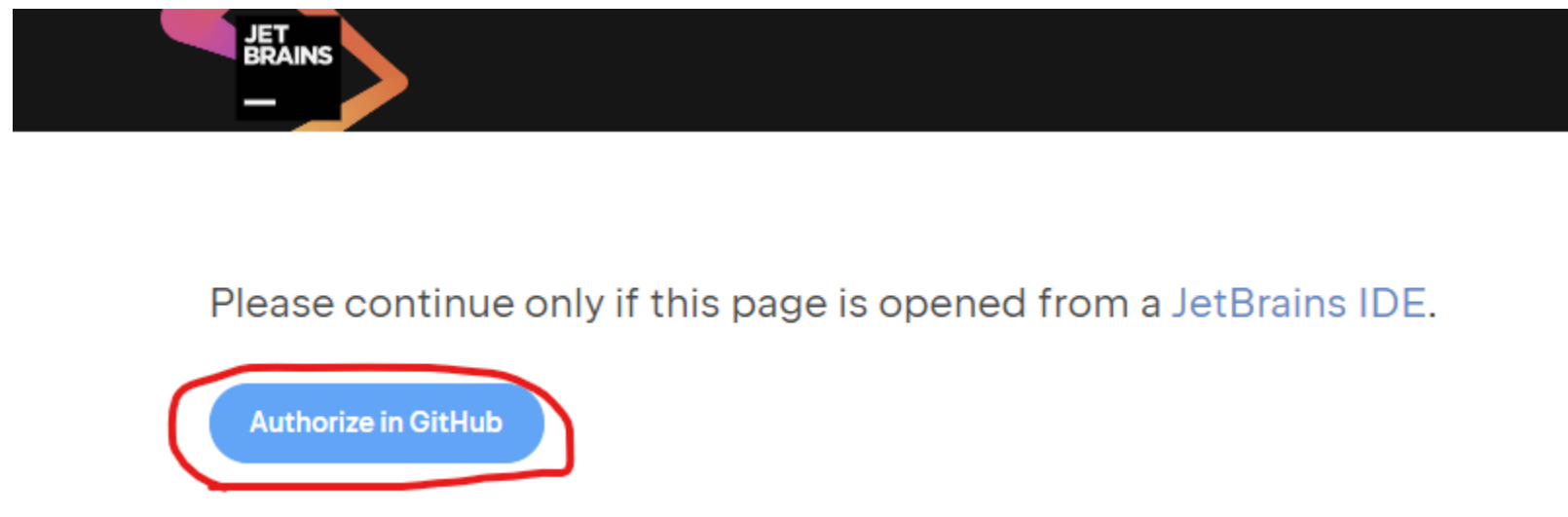
GitHub

IntelliJ GitHub에 로그인하여 내 Repository Clone 하기



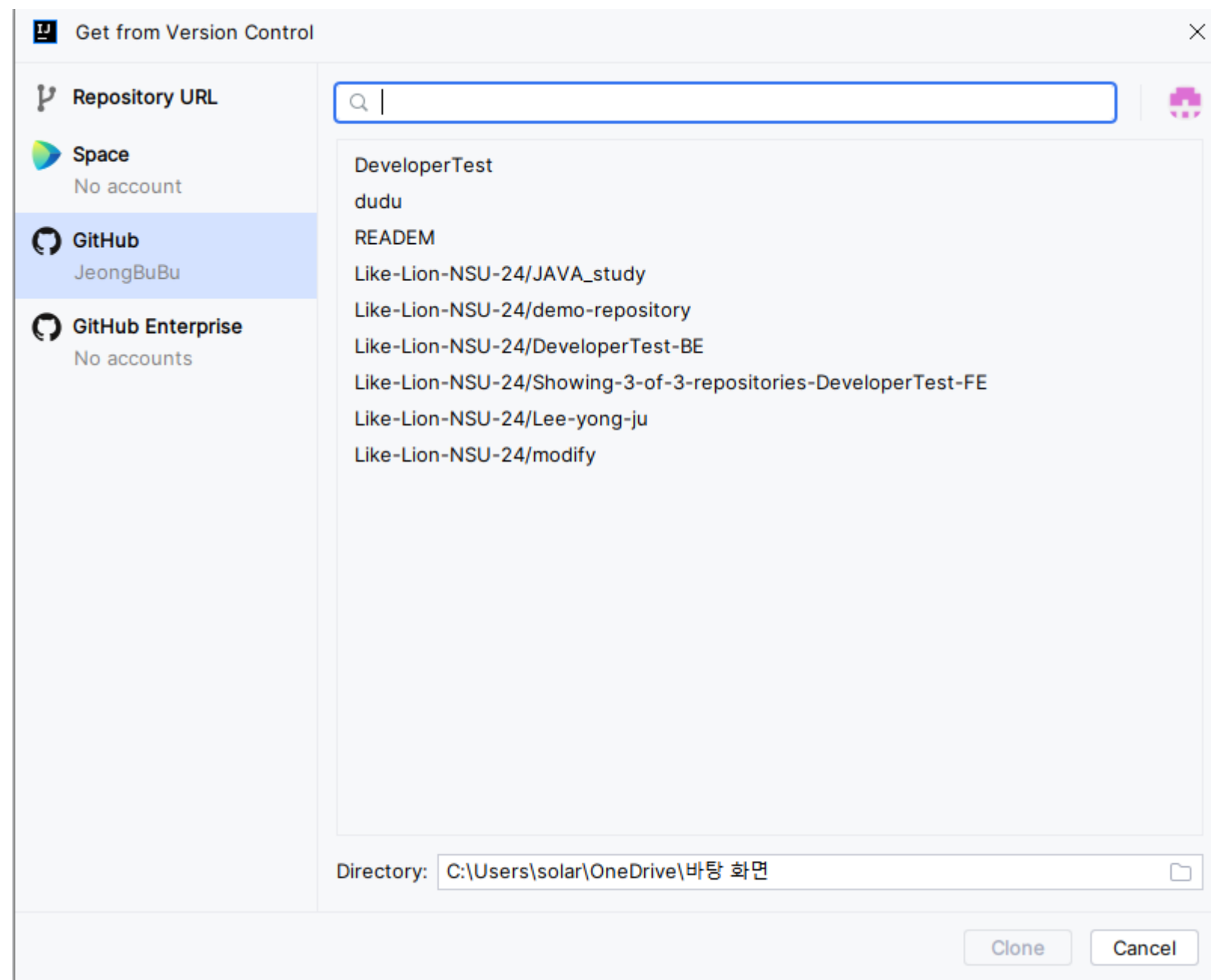
GitHub

IntelliJ GitHub에 로그인하여 내 Repository Clone 하기



GitHub

IntelliJ GitHub에 로그인하여 내 Repository Clone 하기



Repository 선택

인증이 끝나면, IntelliJ에서 내 Repository 목록을 조회할 수 있습니다. Clone 하려는 Repository를 선택하고, Clone 버튼을 클릭합니다.



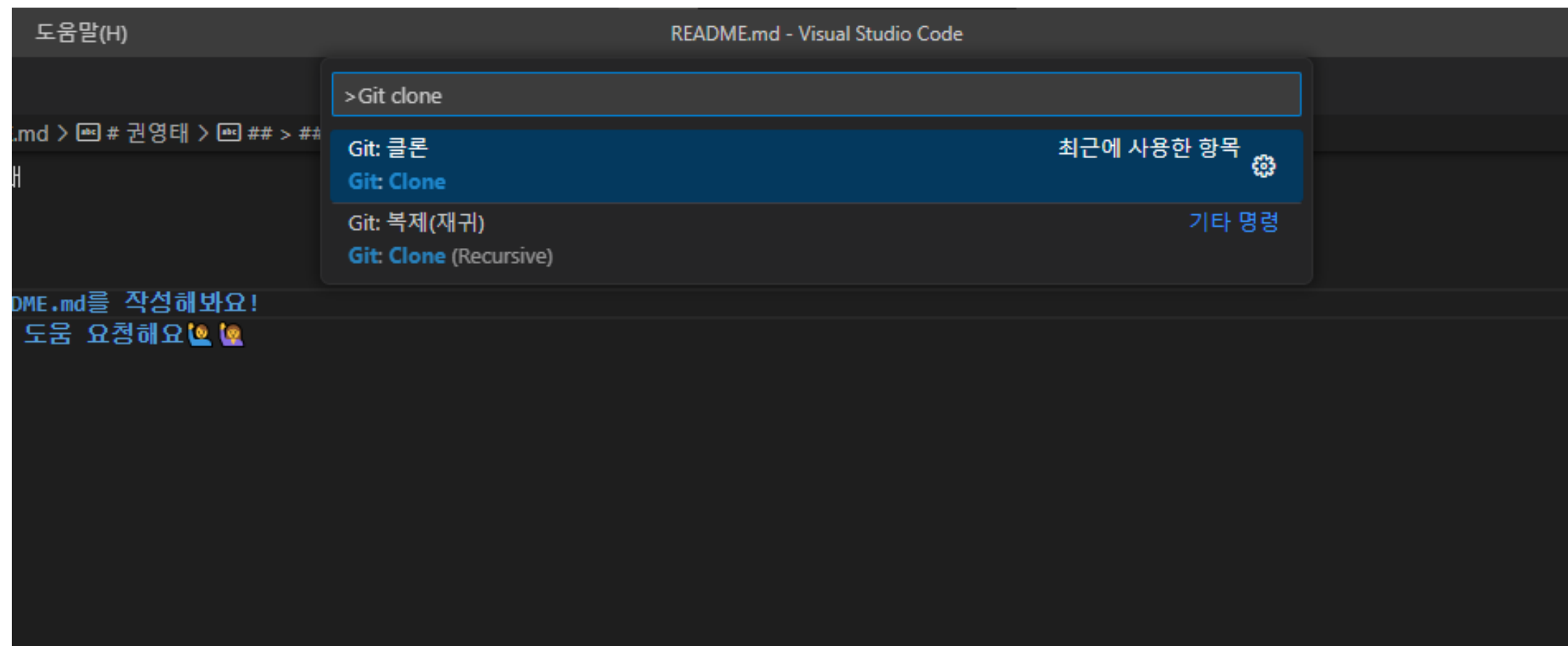
멋쟁이사자처럼
AT 남서울대



05 . VSCode Git 연동

GitHub

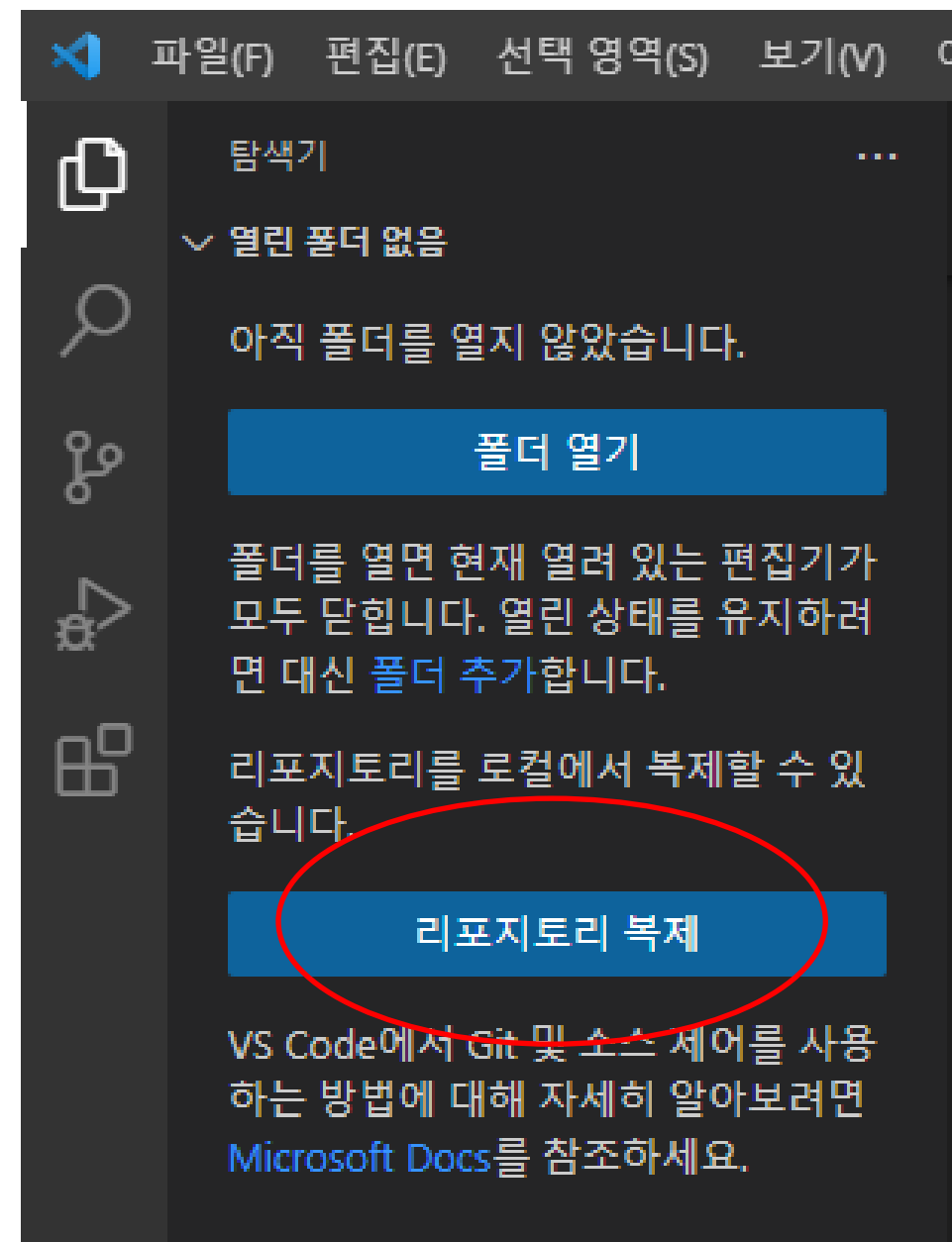
VSCode GitHub 연동





GitHub

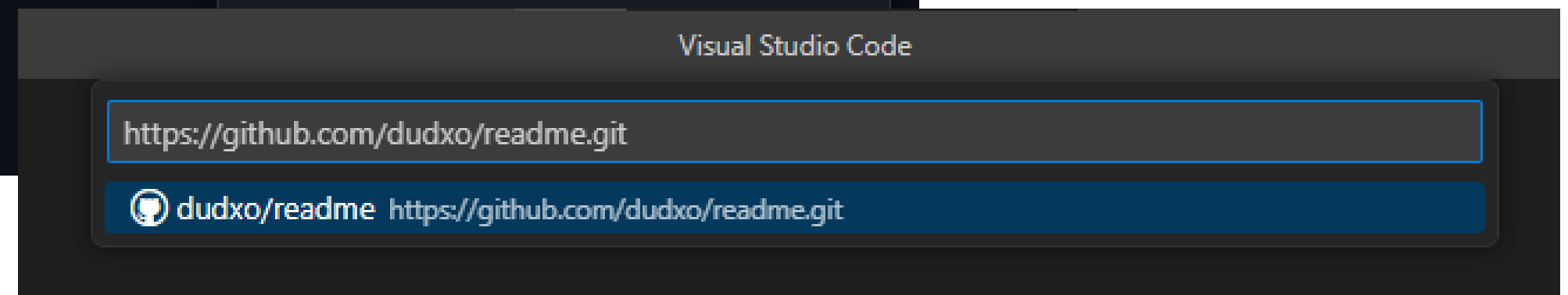
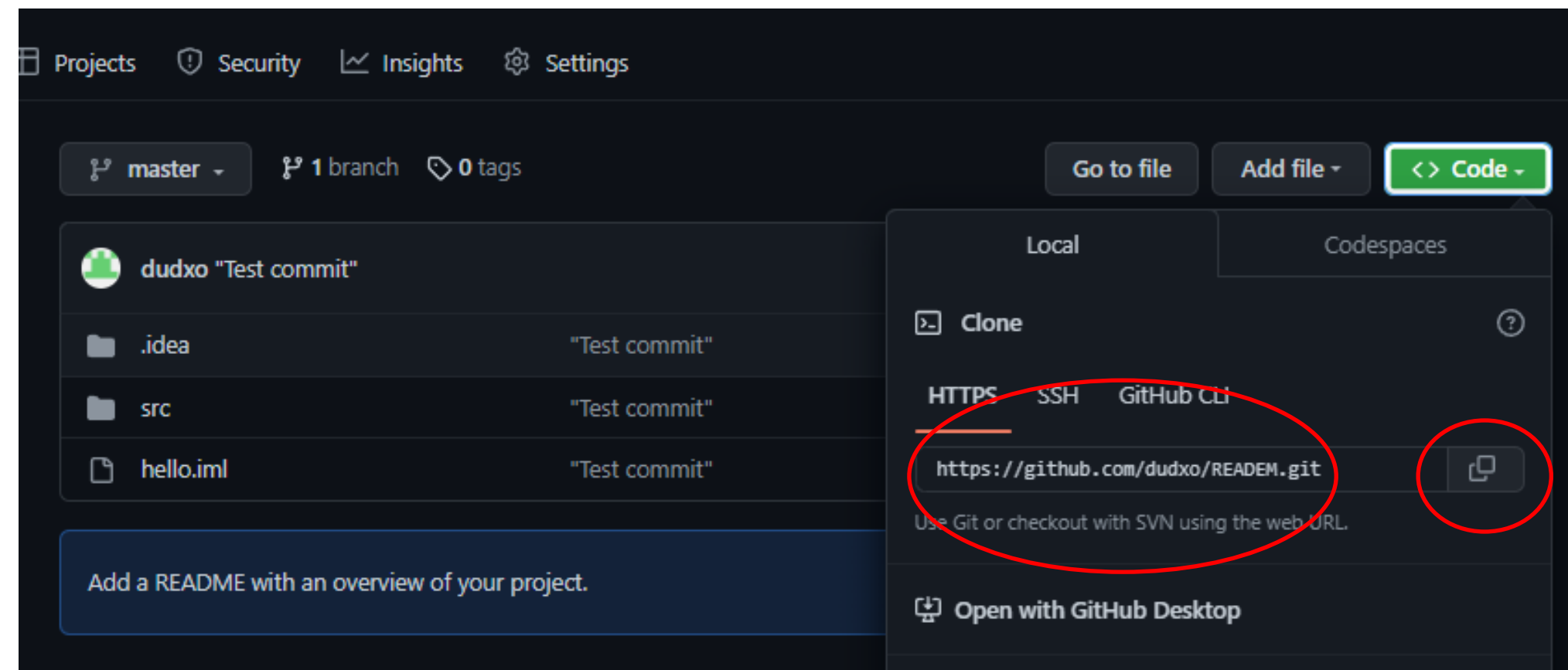
VSCode GitHub 연동





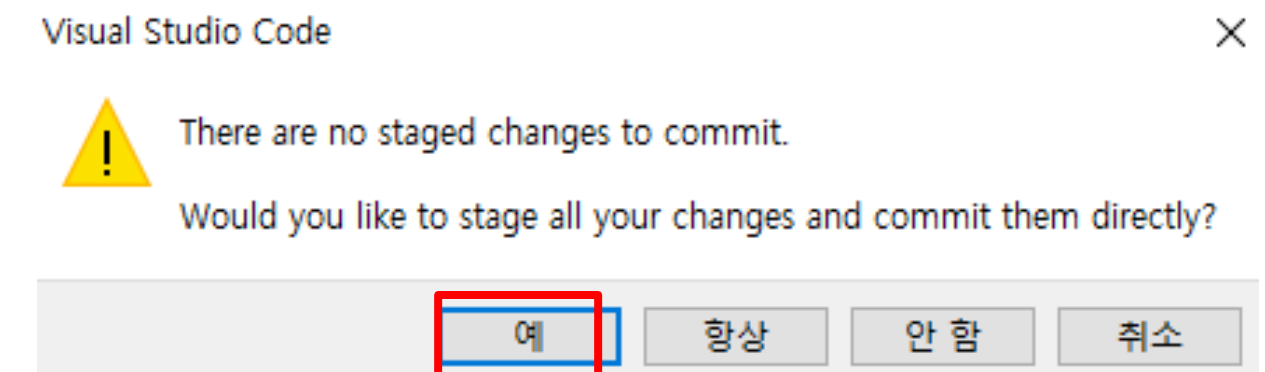
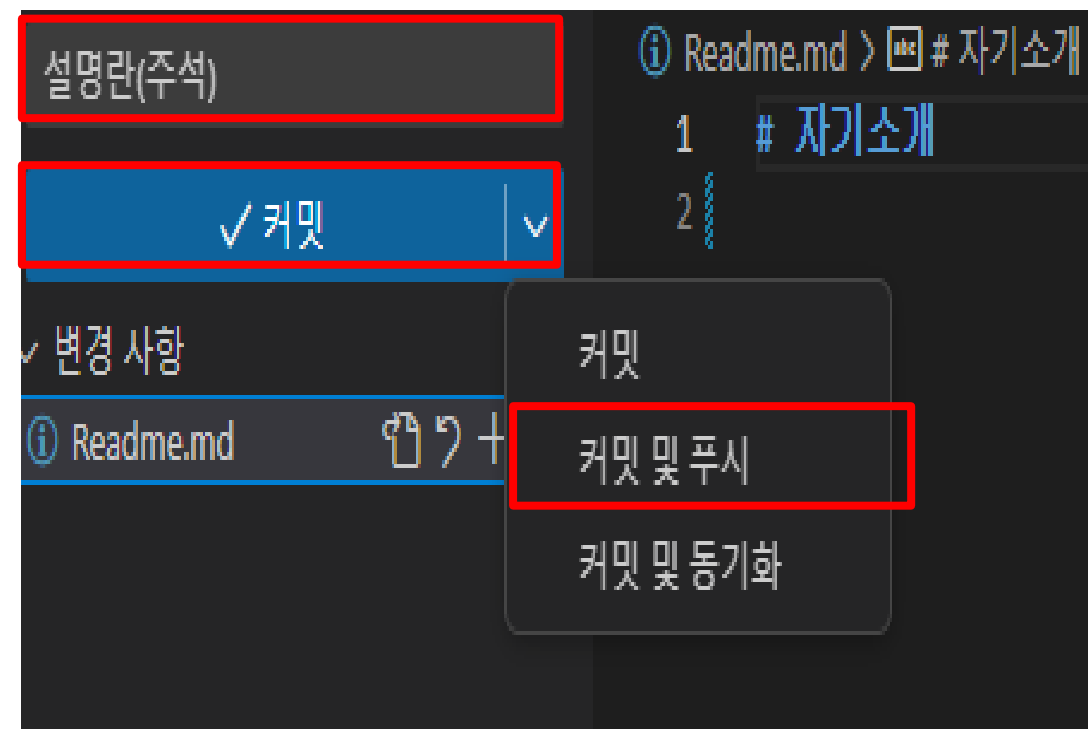
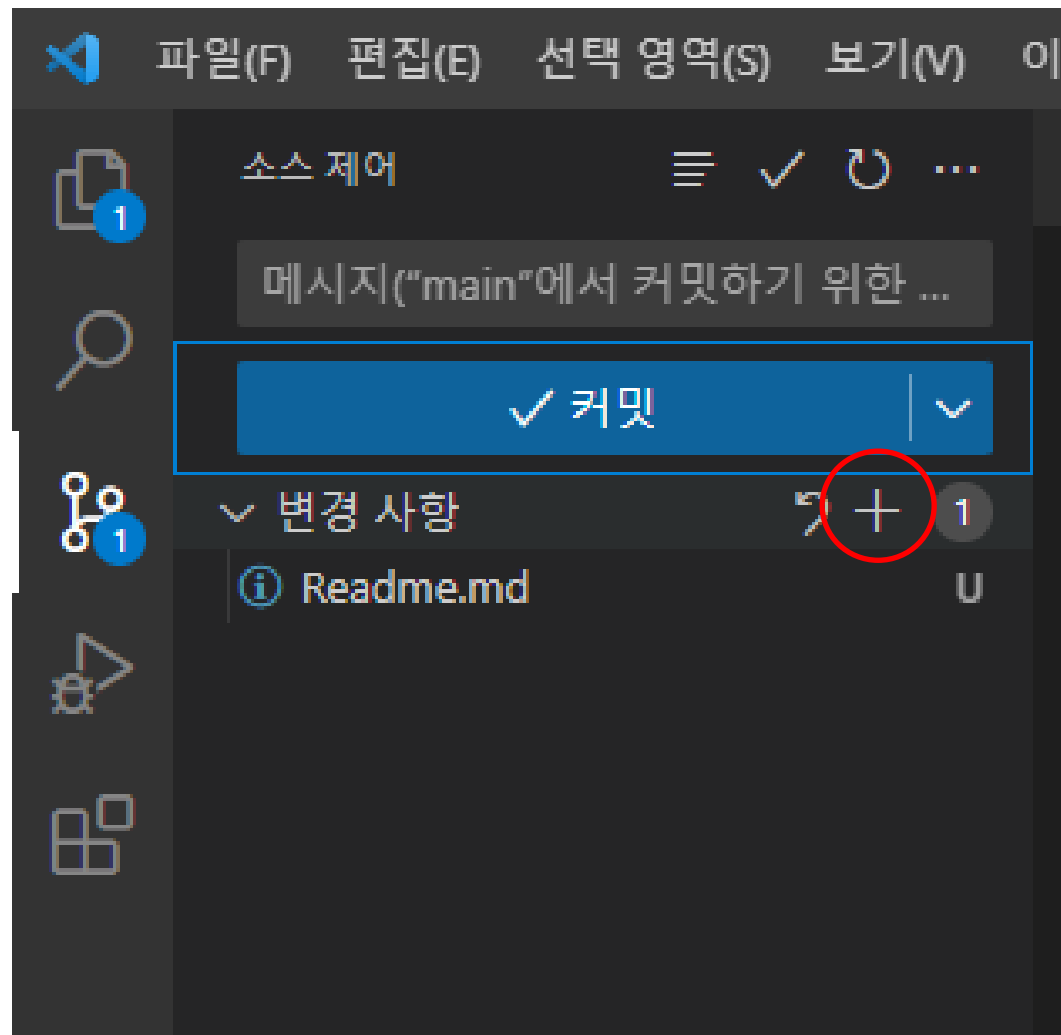
GitHub

VSCode GitHub 연동



GitHub

VSCode GitHub 연동





GitHub

Readme.md(Markdown)

Markdown을 이용해서 자기소개(Readme.md)를 만들어보자!

<https://github.com/JeongBuBu/README>



멋쟁이사자처럼
AT 남서울대



Q&A

발표자: 킹왕짱

전화 ☎ 010-1234-5678

메일 ✉ email@email.com