

# Eclipse – Git 연동

---

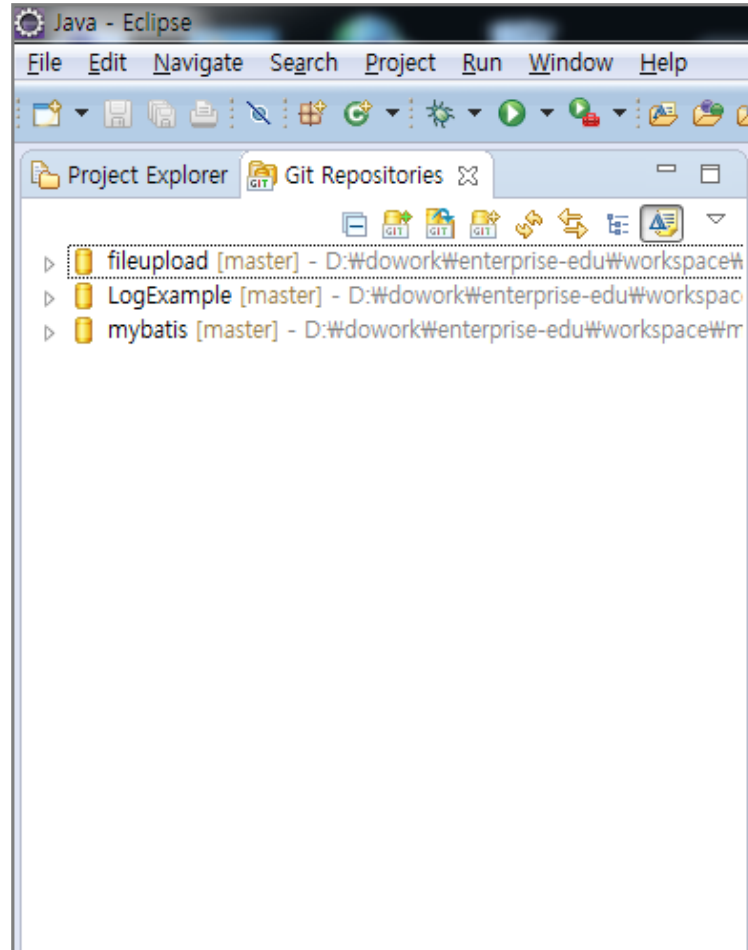
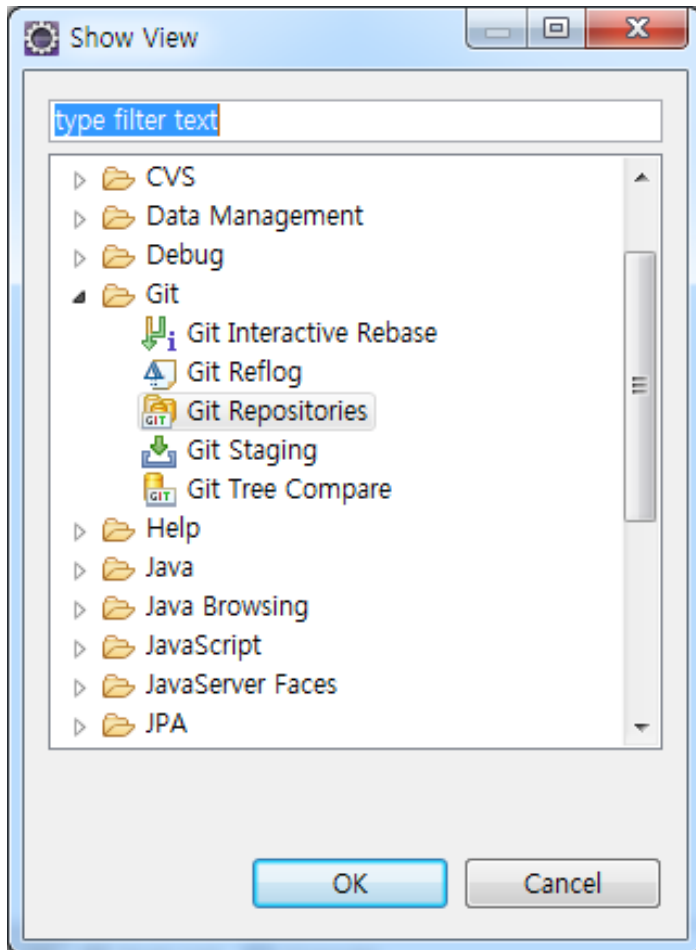
광명융합기술교육원  
데이터분석과

# 기존 프로젝트 gitHub에 올리기

1. Remote와 연결
2. Commit & push
3. Pull
4. Branch checkout

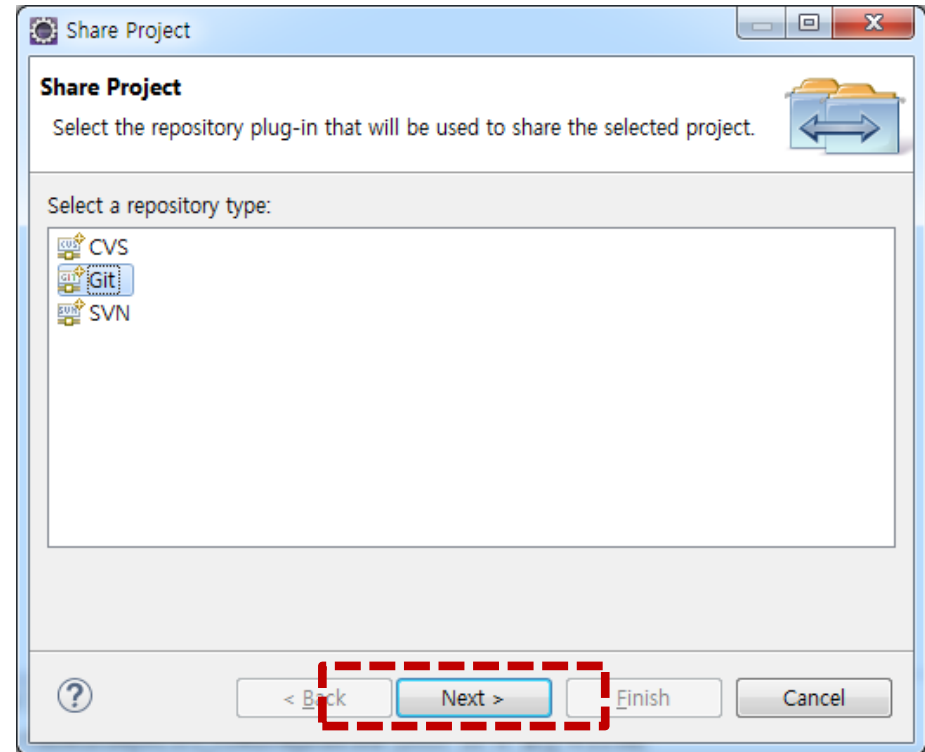
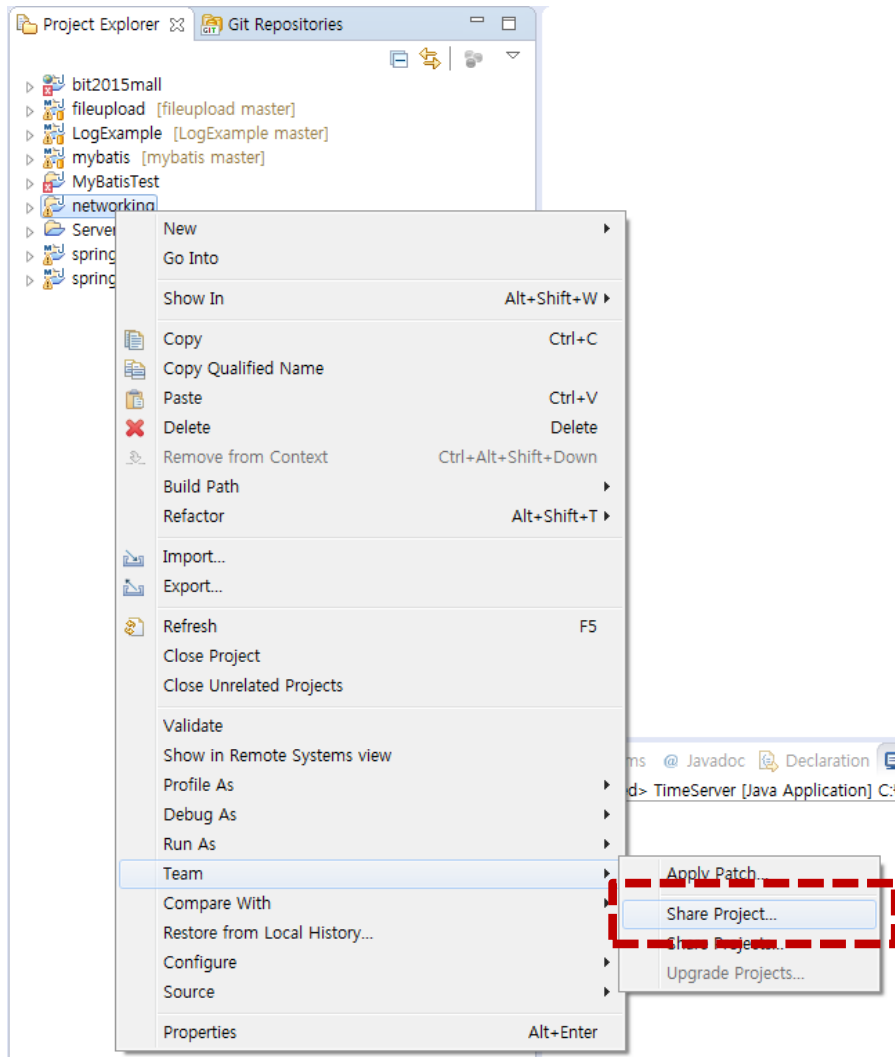
# Git Repositories 뷰 꺼내오기

- Window > Show View > Other



# 기존 프로젝트에서 Team-Share Project

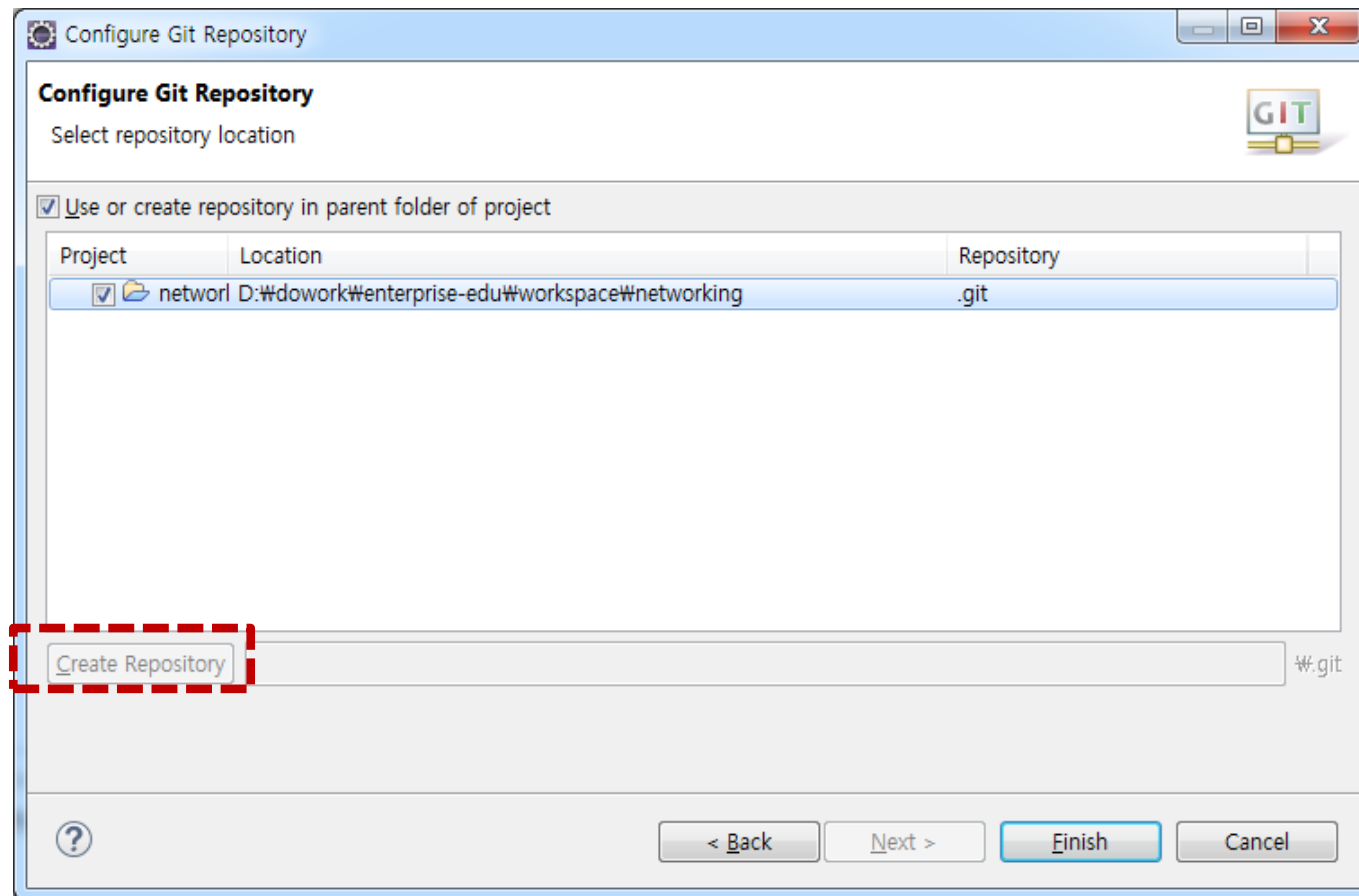
Remote 연결



# Local git repository 설정

Remote 연결

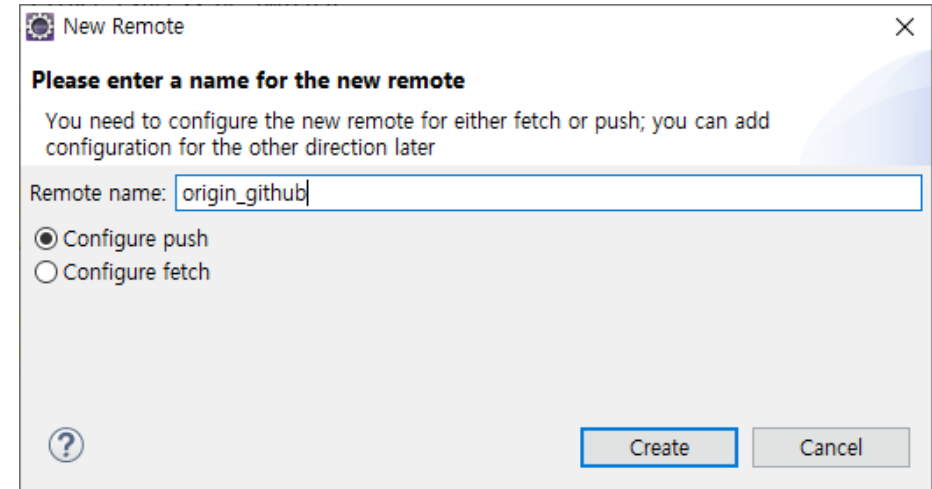
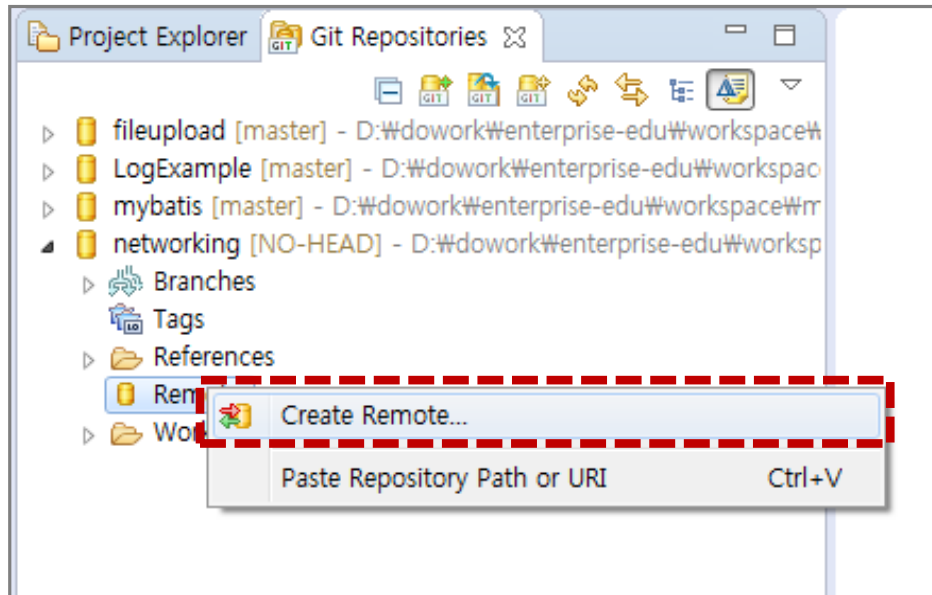
- 따로 로컬 레포지토리를 생성하지 않고 이클립스 프로젝트를 local git repository로 사용



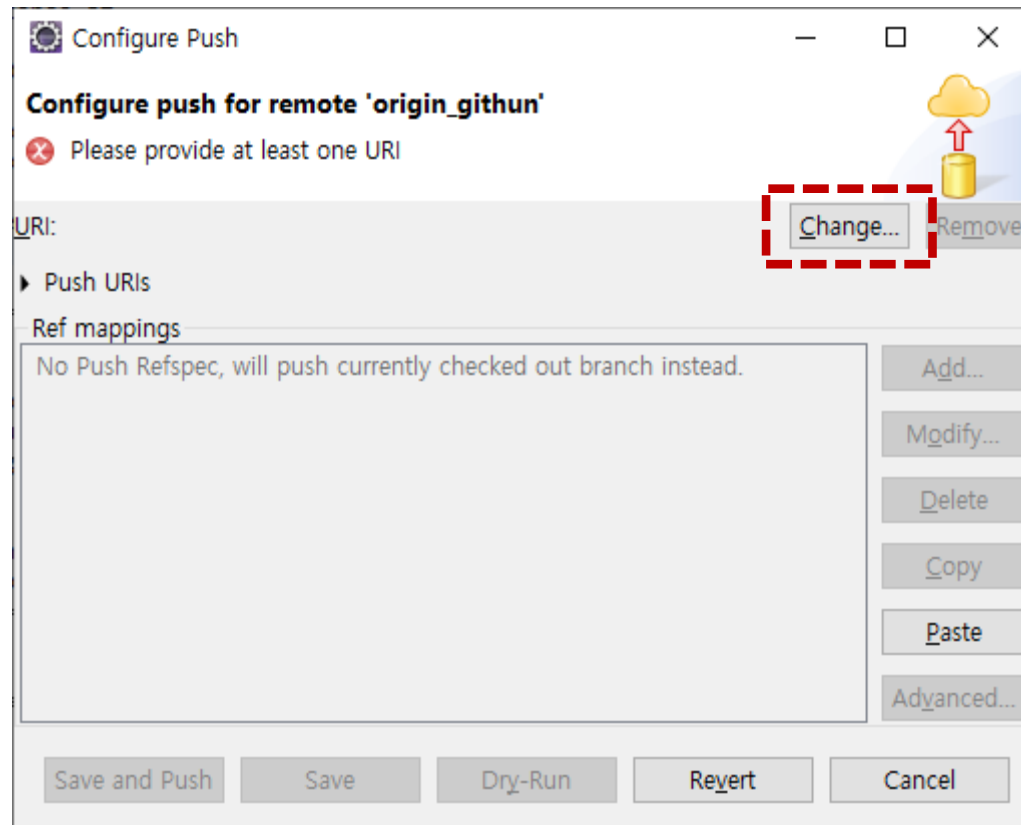
# 원격 레포지토리 생성 (1/3)

Remote 연결

- 원격 레포지토리를 (github.com) 과 연결한다.
- Remote (원격 레포지토리)의 이름을 정한다. (default- origin)



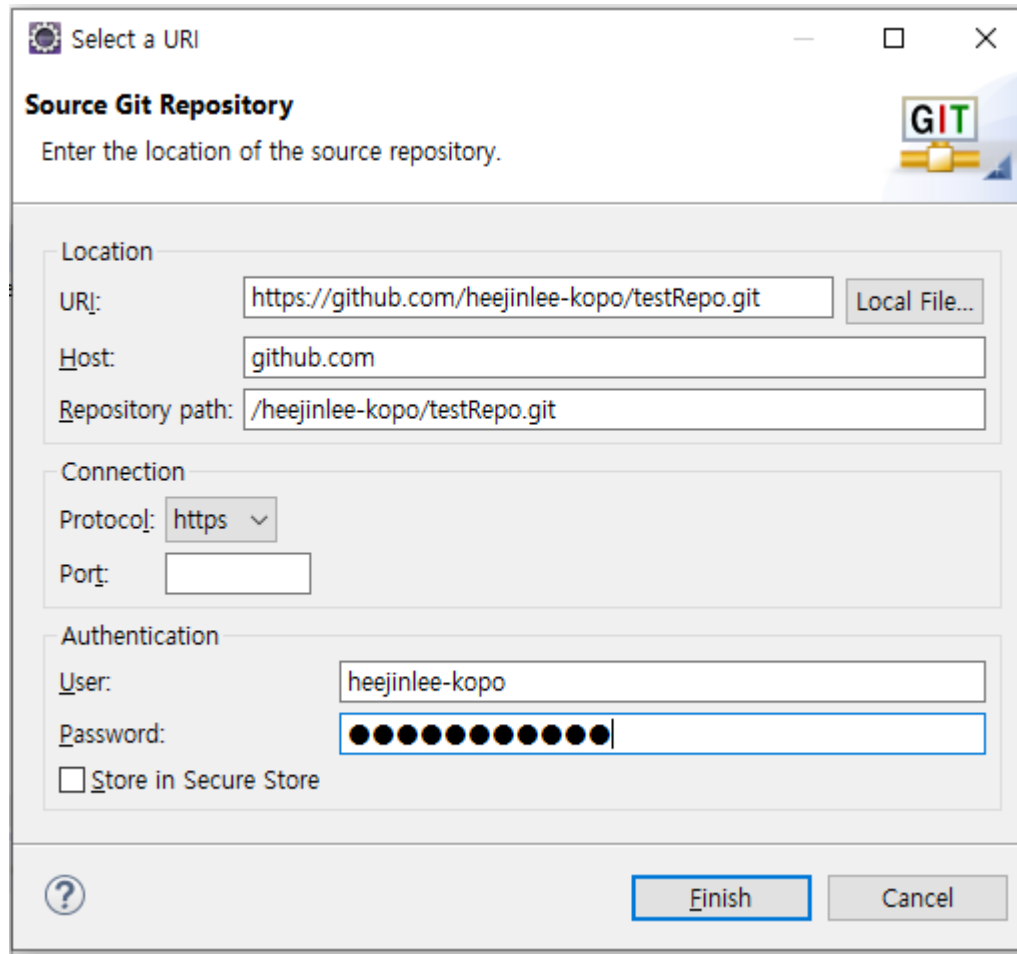
- Origin이 가리키는 원격 정보 주소를 넣는다.
  - Github의 repository – <http://github.com/heejinlee-kopo/testRepo.git>
  - change를 누르기 전에 <http://github.com> 에서 새 레포지토리를 만든다.



# 원격 레포지토리 생성 (3/3)

Remote 연결

- 미리 github에 repository가 존재해야 함



The screenshot shows a Windows-style dialog box titled "Select a URI". Inside, there's a section "Source Git Repository" with a Git logo and the instruction "Enter the location of the source repository." The dialog is divided into three main sections: "Location", "Connection", and "Authentication".

**Location:**

- URI:
- Host:
- Repository path:

**Connection:**

- Protocol:  (dropdown arrow)
- Port:

**Authentication:**

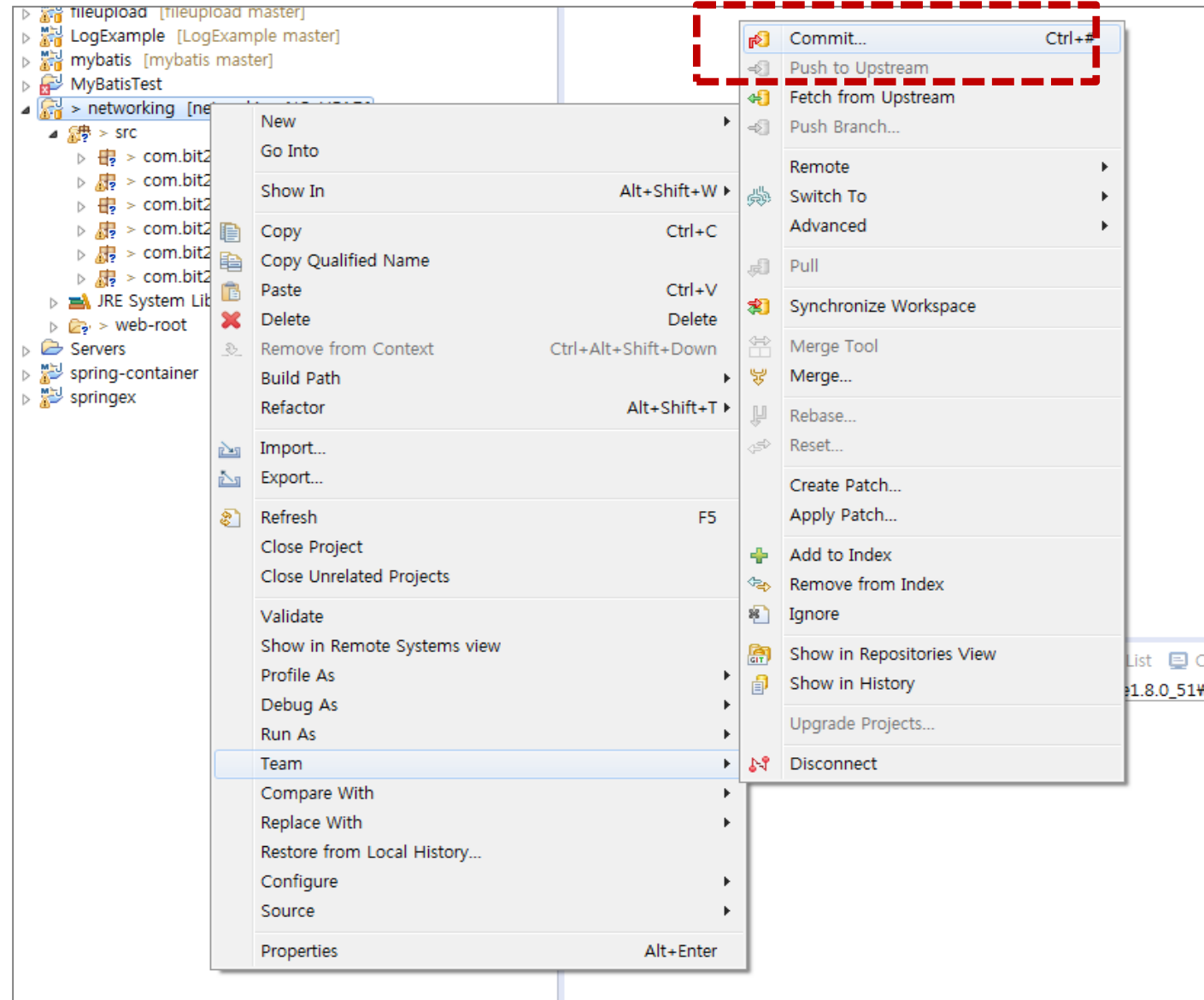
- User:
- Password:
- ☐ Store in Secure Store

At the bottom, there's a help icon (question mark), an "Finish" button, and a "Cancel" button.



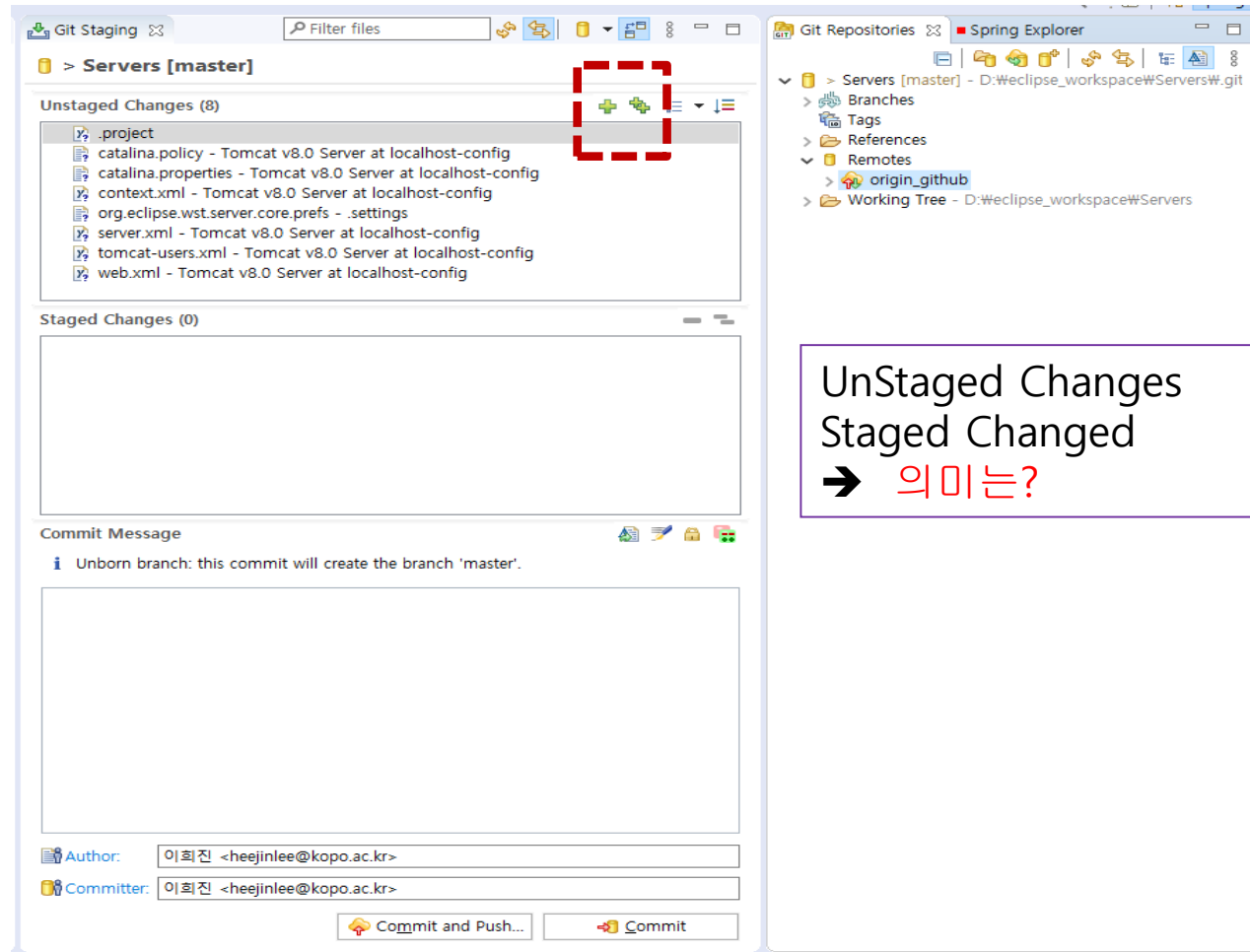
# Commit and push

- 기존 code를 commit
- Team-Commit



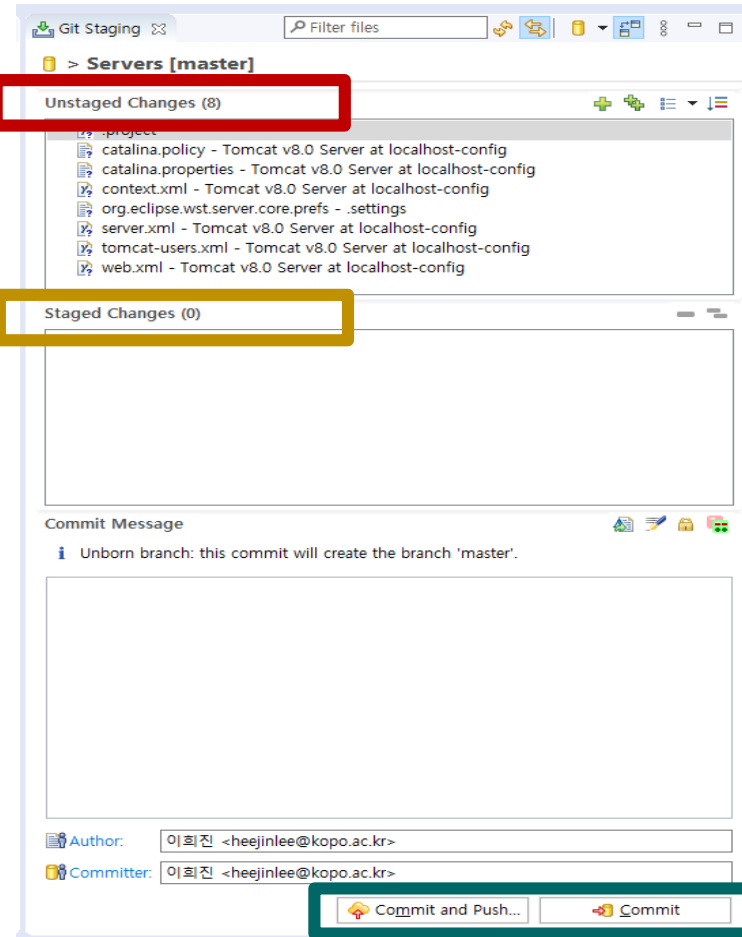
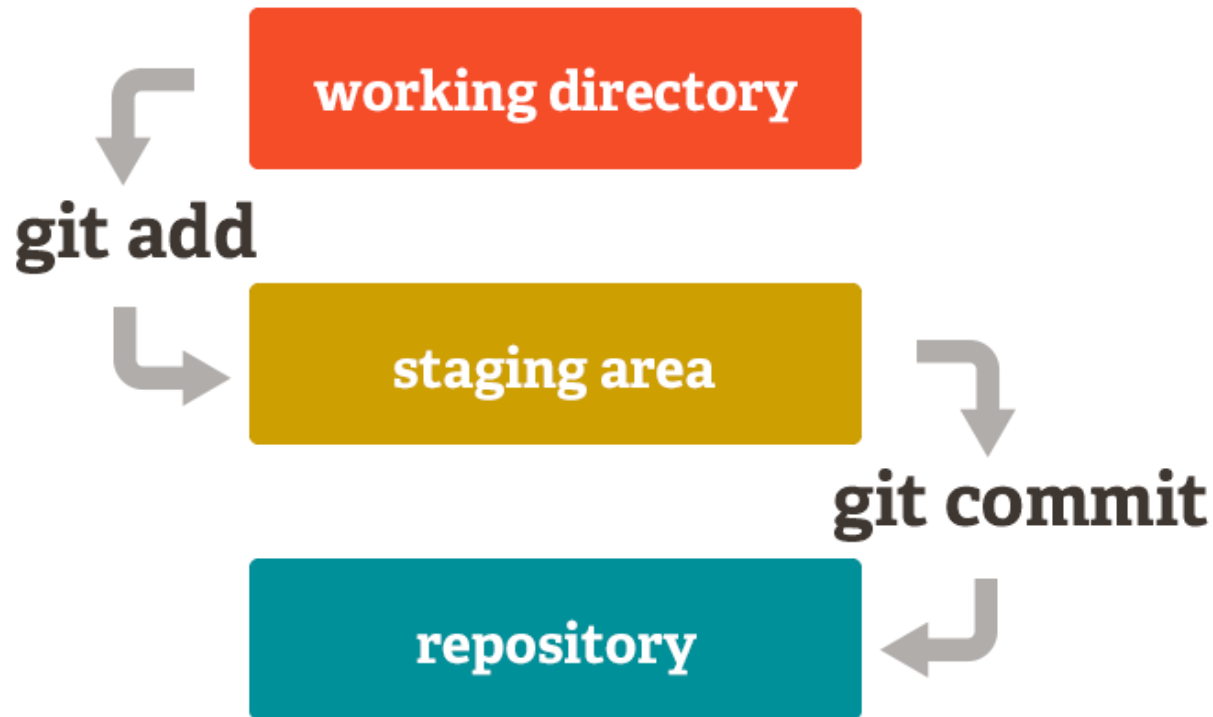
# Unstaged/Staged Changes (1/2)

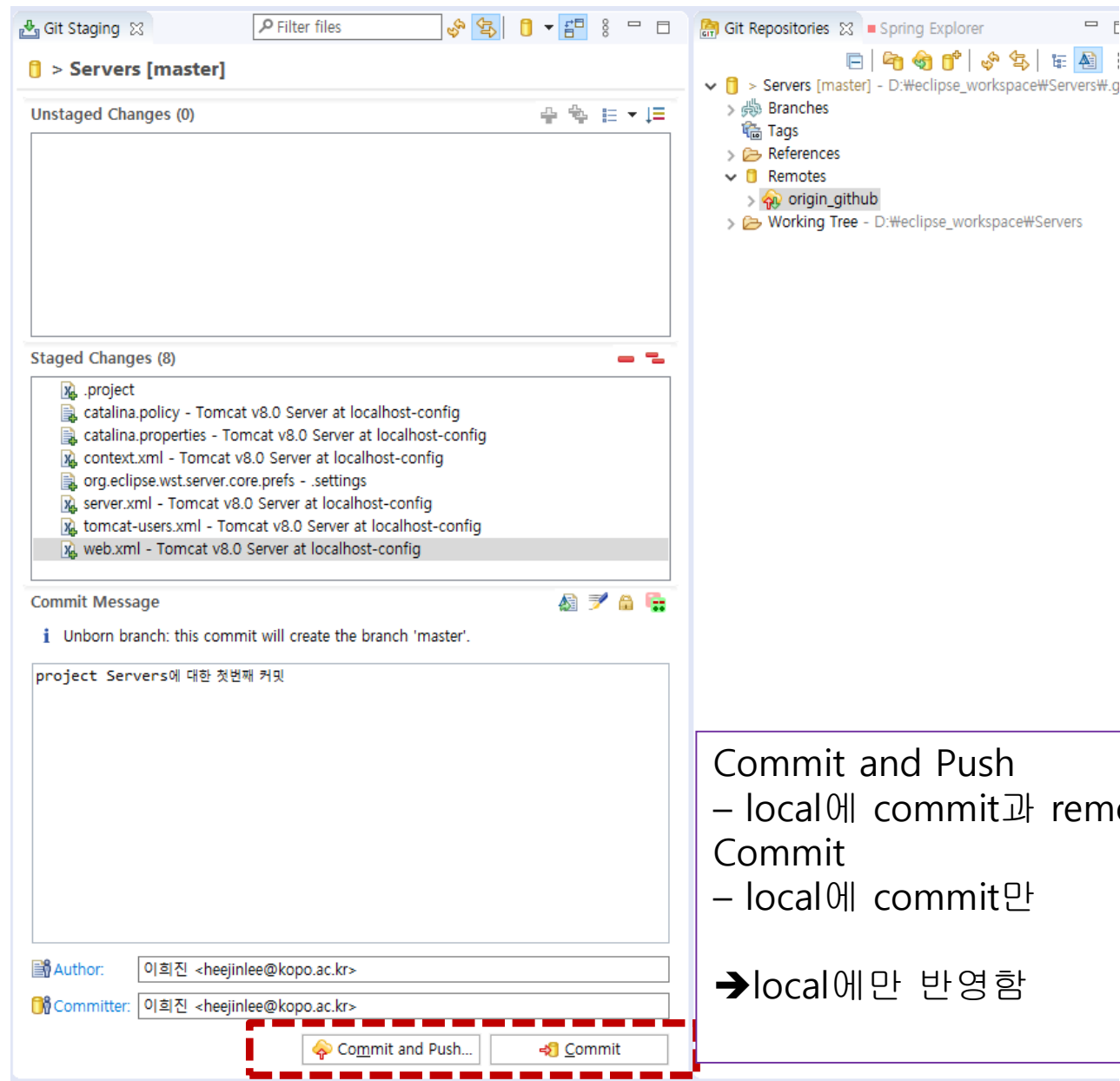
- Window "Git Staging"
- Window "Git Repositories"



# Unstaged/Staged Changes (2/2)

Commit & Push

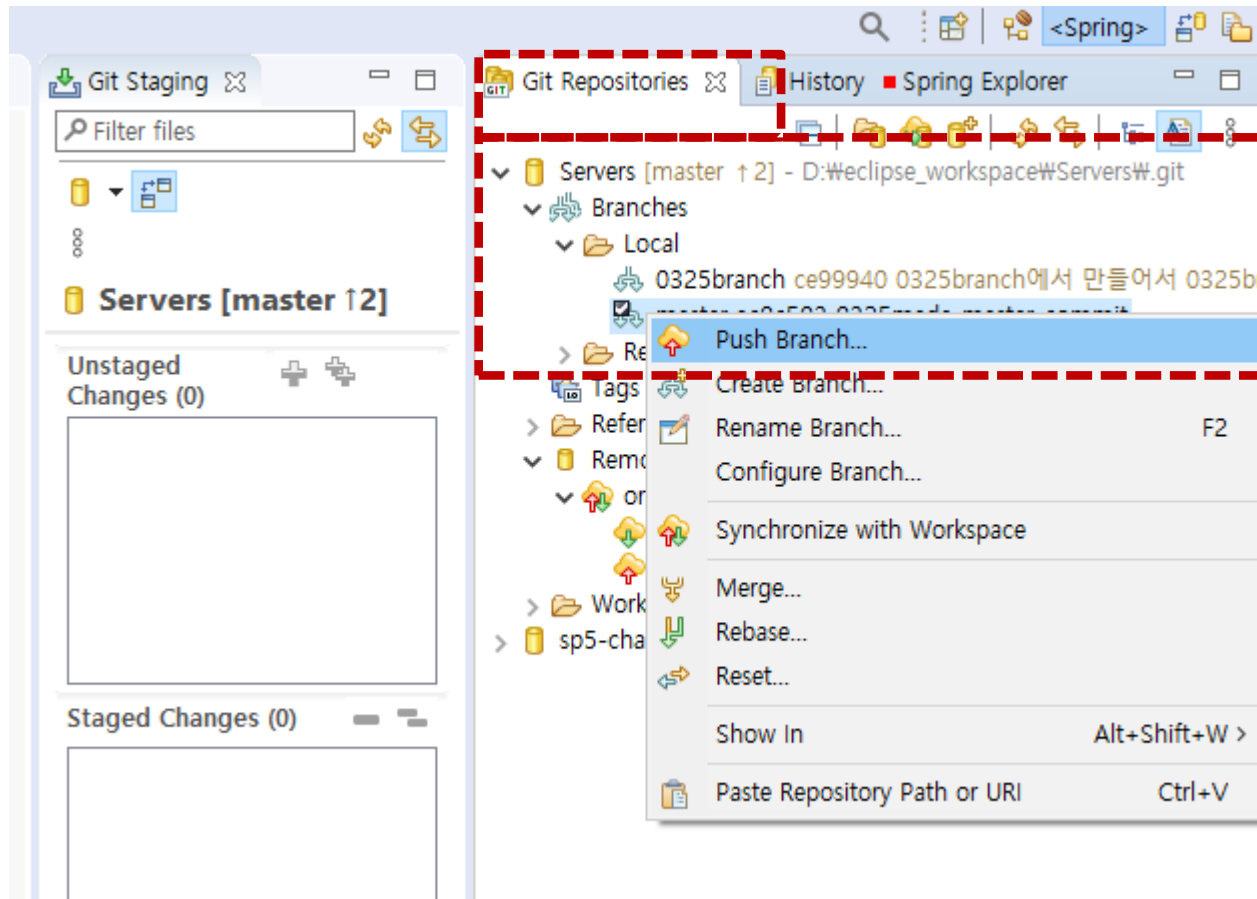




### Commit and Push

- local에 commit과 remote에 push를 동시에 Commit
- local에 commit만

➔local에만 반영함



# Push – remote 에 반영

Commit & Push

Push Branch master

**Push to branch in remote**

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

Source:

master 6a20a64 project Servers에 대한 첫번째 커밋

Destination:

Remote: origin\_github: https://github.com/heejinlee-kopo/testRepo.git

Branch: master

☒ Configure upstream for push and pull

When pulling: Merge

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

Show [advanced push](#) dialog

< Back Preview > Push Cancel

Remote에 반영하자!!

Remote

- 반영할 외부의 github repository

Branch

- 반영할 외부 github의 repository의 branch

# Push confirmation에서 에러가 난다. 왜지

Commit & Push

## [rejected – non-fast-forward] 에러

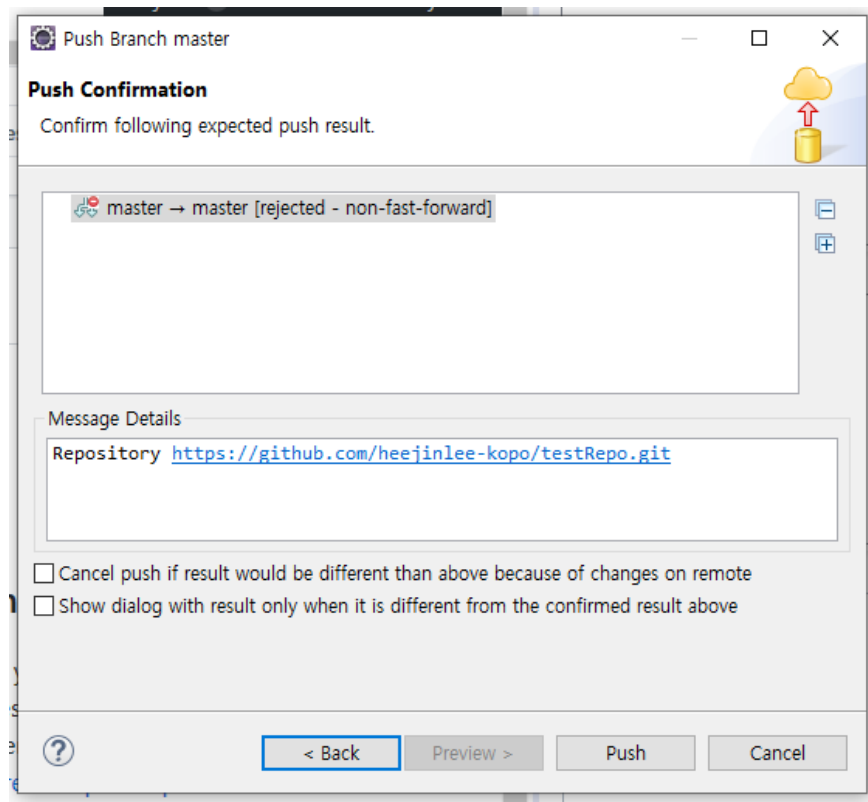
### • 원인

- 깃 허브에 생성된 원격 저장소와 로컬에 생성된 저장소 간 공통분모가 없는 상태에서 병합 하려는 시도로 인해 발생
- 기본적으로 관련 없는 두 저장소를 병합하는 것은 안되도록 설정이 되어 있기 때문에

### • 해결

- 두 저장소를 병합하도록 허용
- Git pull시에 `--allow-unrelated-histories` 옵션 추가하여 두 저장소 병합

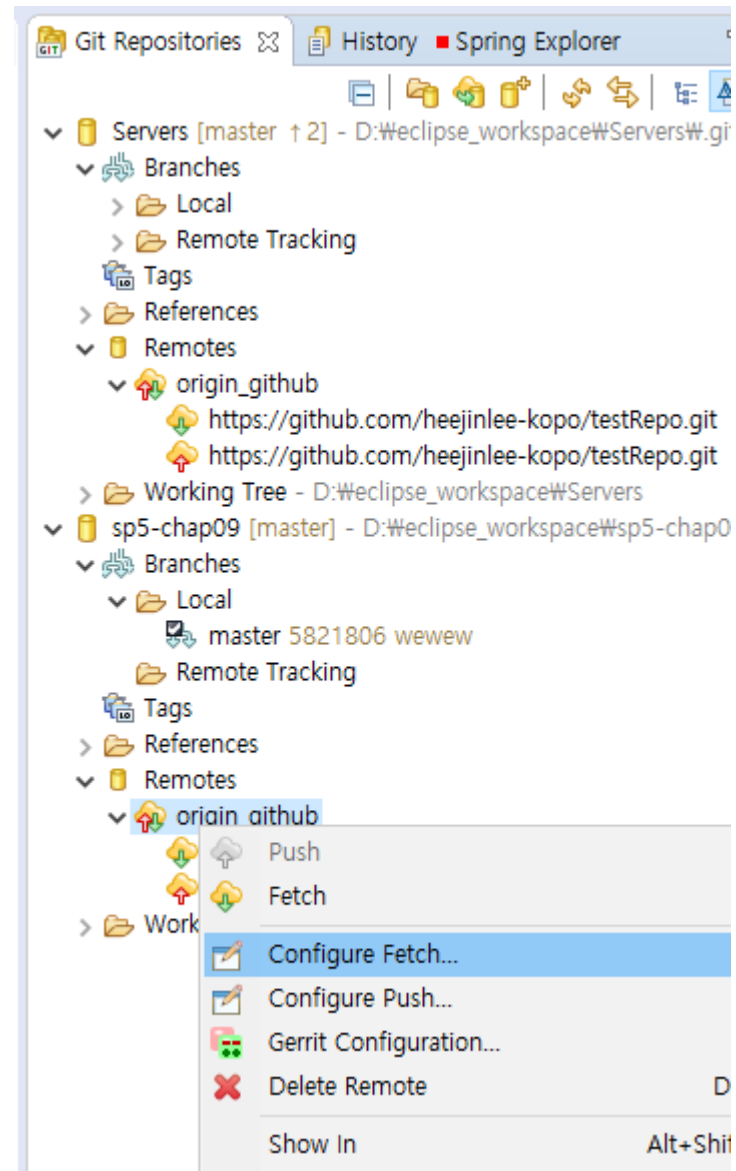
```
git pull origin master --allow-unrelated-histories
```



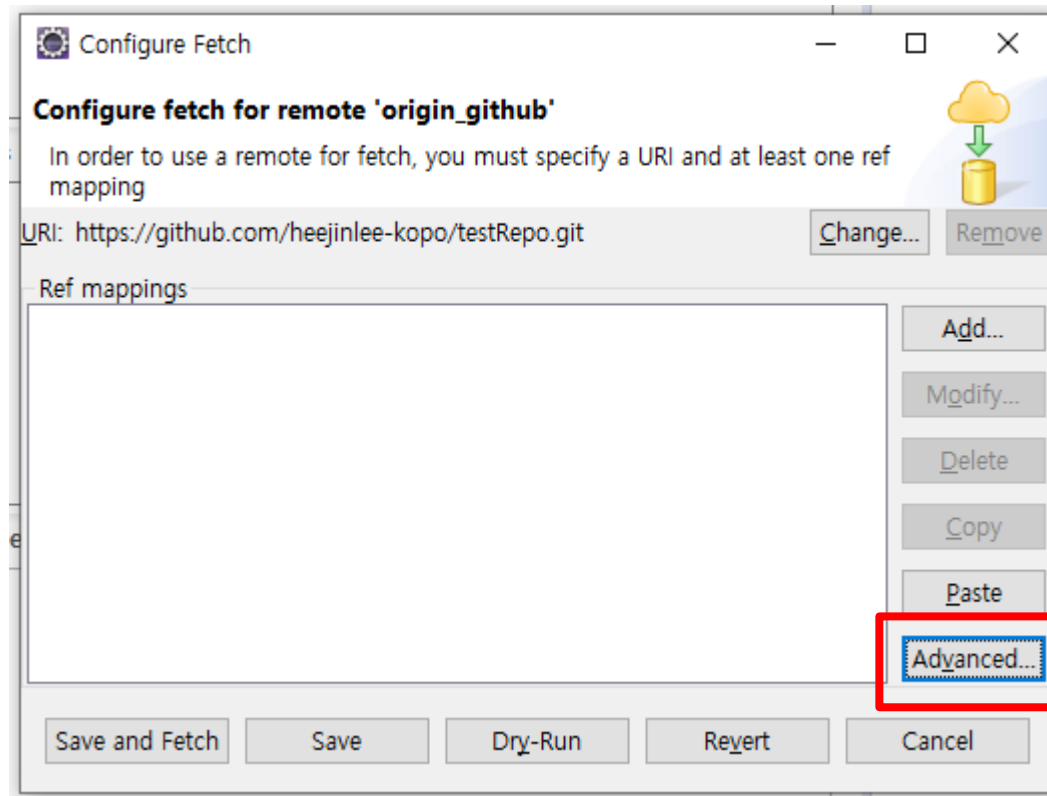
# 문제를 해결하자- configure fetch (1/5)

Push confirmation error

- Window "Git Repositories"
  - Remotes->origin\_github-> Configure Fetch







# 문제를 해결하자- configure fetch (3/5)

Push confirmation error

Source ref:  
Refs/heads/master

Destination ref:  
refs/remotes/"외부\_repository\_이름"/master

**Configure Fetch**

**Fetch Ref Specifications**  
Select refs to fetch.

Add create/update specification  
Source ref:  Destination ref:

Add predefined specification

Specifications for fetch

Source Ref	Destination Ref	Force Update	Remove

Annotated tags fetching strategy  
☒ Automatically follow tags if we fetch the thing they point at  
☐ Always fetch tags, even if we do not have the thing it points at  
☐ Never fetch tags, even if we have the thing it points at

☐ Save specifications in 'origin\_github' configuration

**Configure Fetch**

**Fetch Ref Specifications**  
Select refs to fetch.

Add create/update specification  
Source ref:  Destination ref:

Add predefined specification

Specifications for fetch

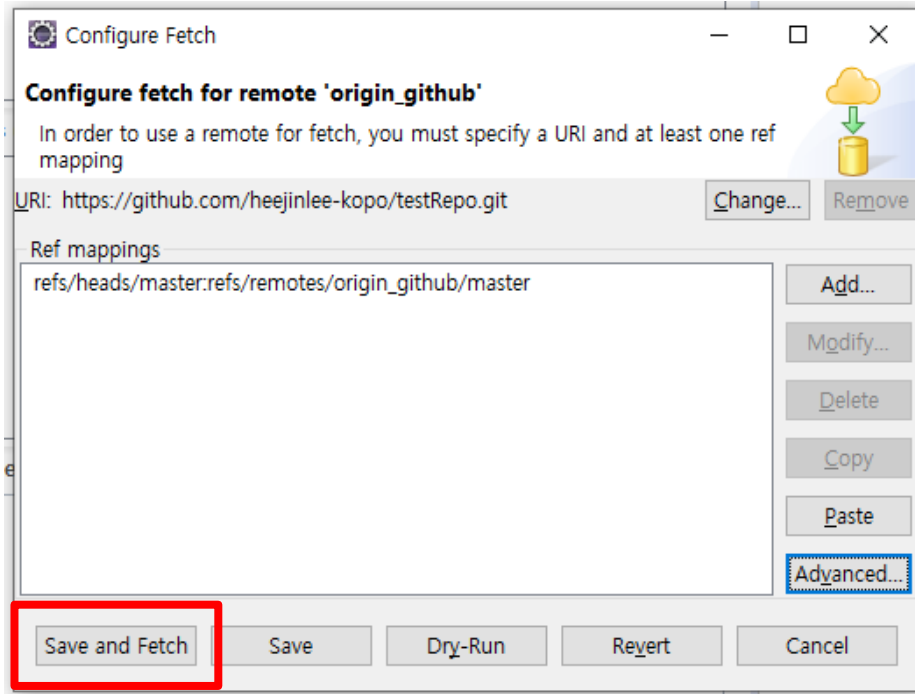
Source Ref	Destination Ref	Force Update	Remove
refs/heads/master	refs/remotes/origin_github/...	<input type="checkbox"/>	

Annotated tags fetching strategy  
☒ Automatically follow tags if we fetch the thing they point at  
☐ Always fetch tags, even if we do not have the thing it points at  
☐ Never fetch tags, even if we have the thing it points at

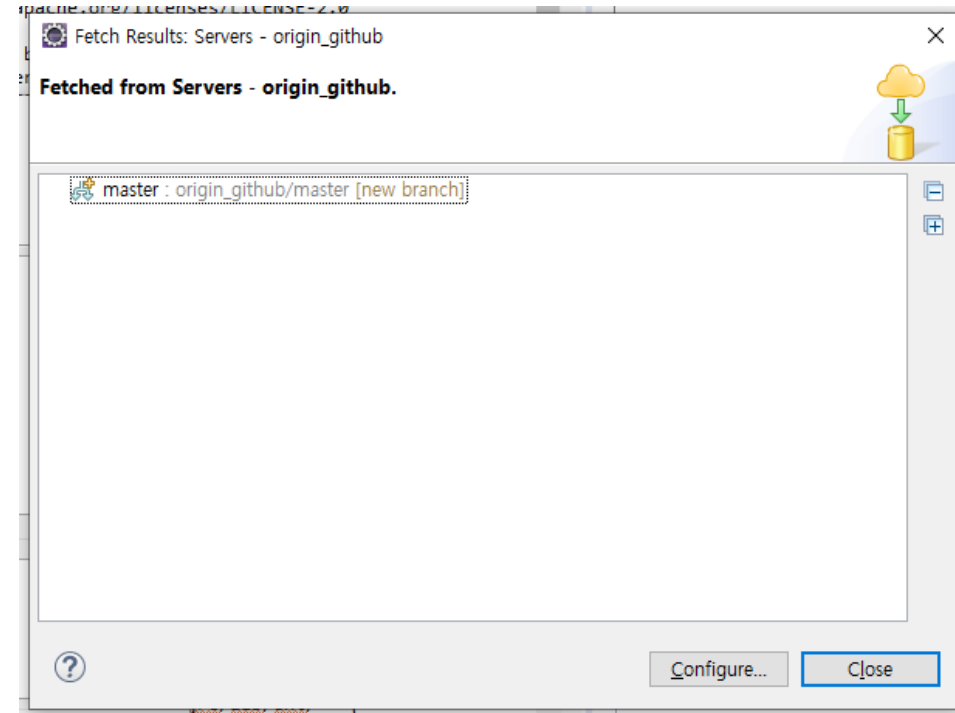
☐ Save specifications in 'origin\_github' configuration

# 문제를 해결하자- configure fetch (5/5)

Push confirmation error

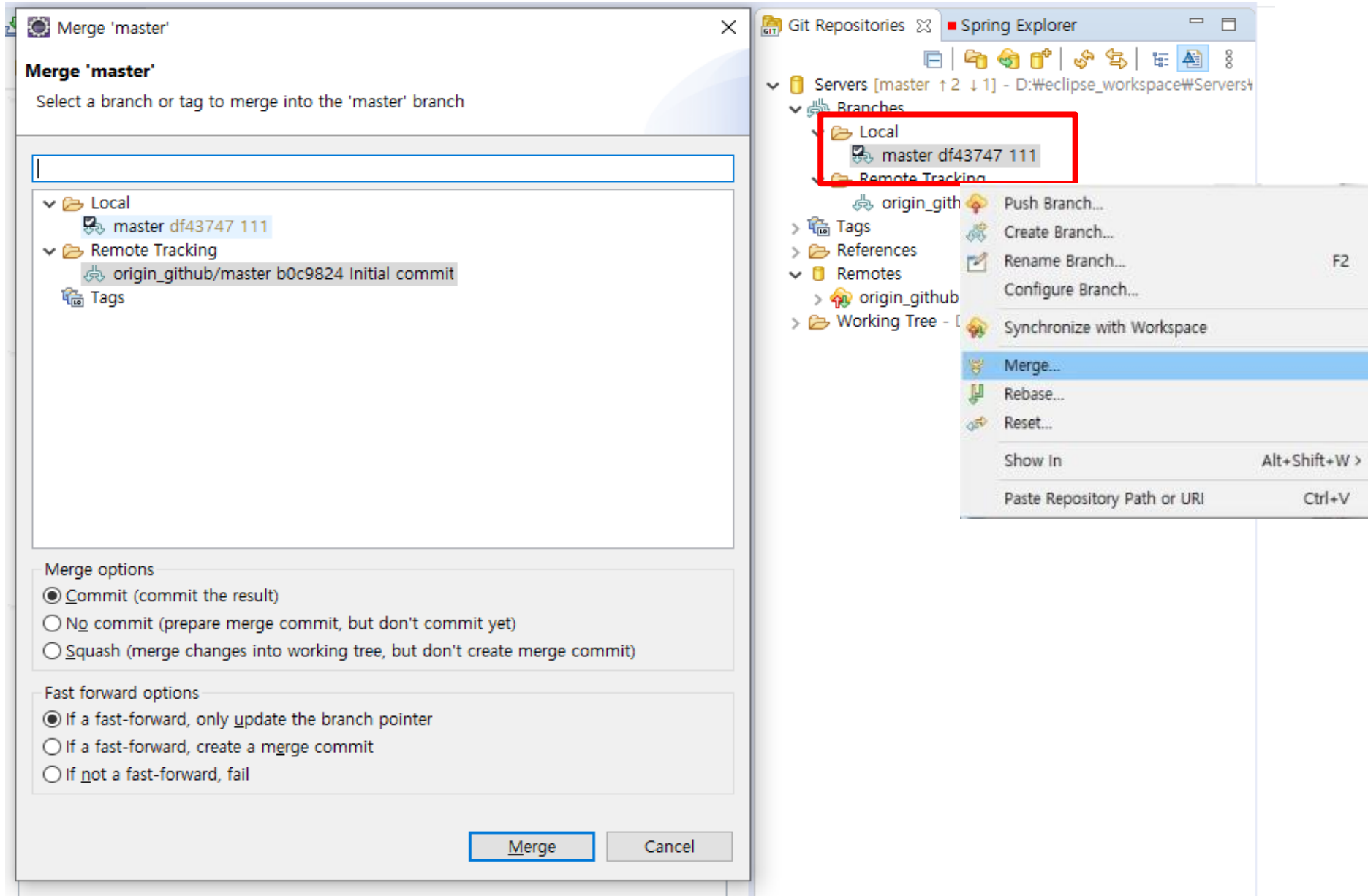


Save and Fetch



# Fetch한 후에 merge해야 pull 완성

Merge



## 지금까지 pull을 했으니, 이제 push하자. - Commit 다시 한번

---

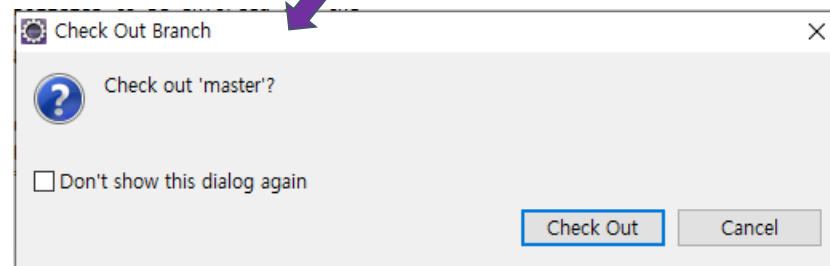
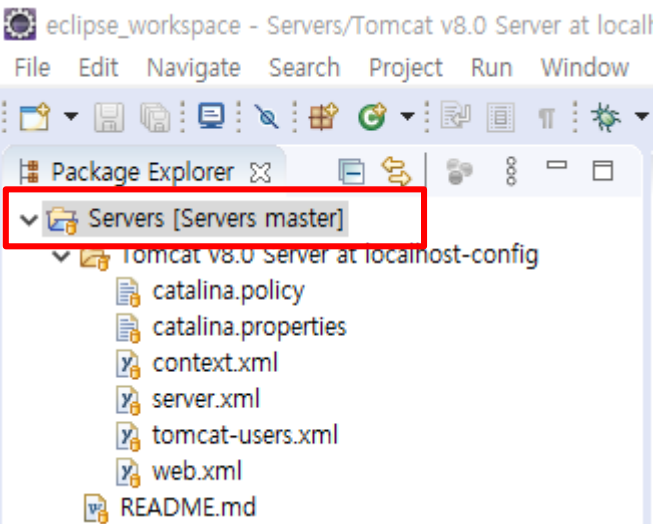
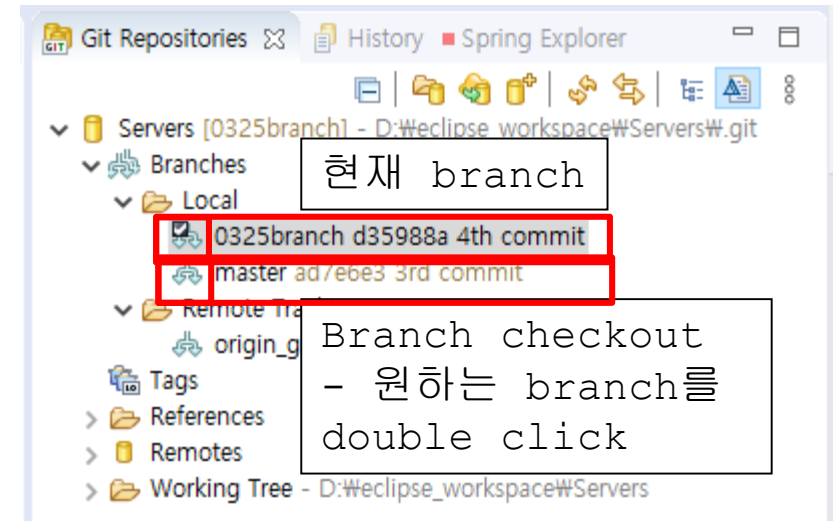
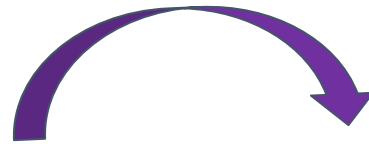
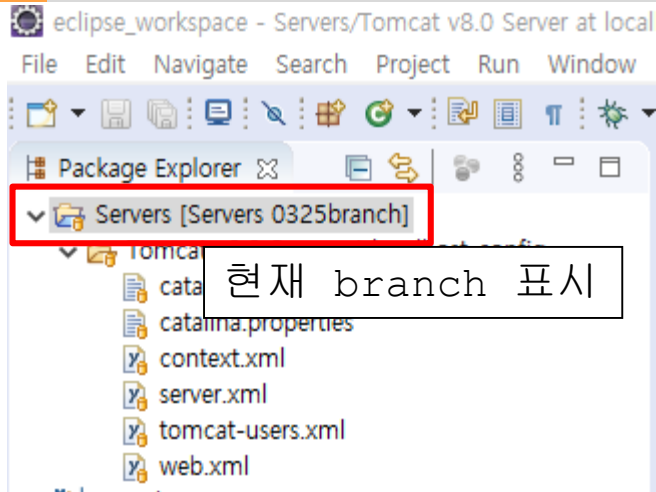
- Merge result ok 나온다.
- 이제 pull을 다 했음.
- 이제 push할 준비가 다 되었으니, push를 해보자.



Branch 변경

# Branch check out – 방법1

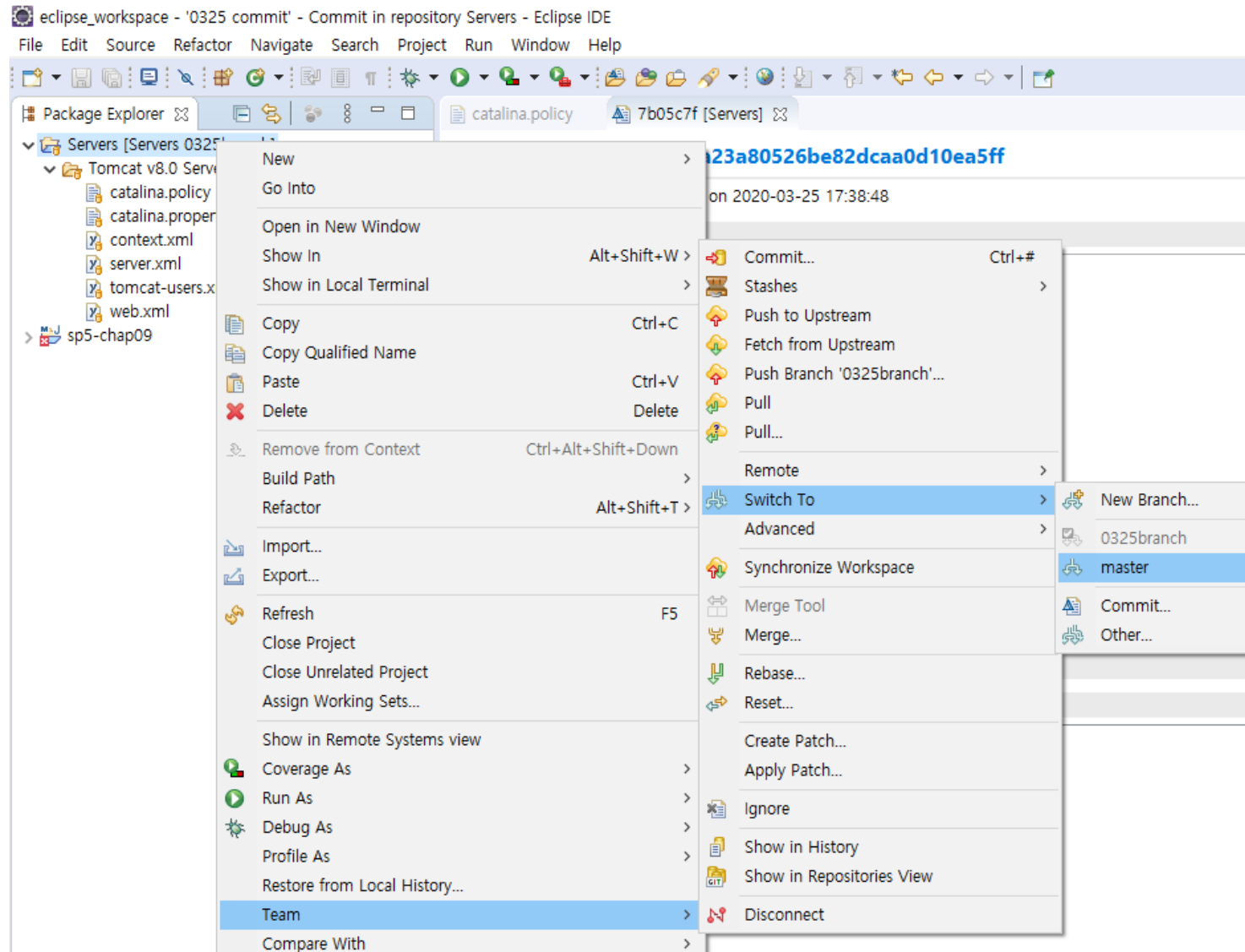
Branch checkout





# Branch check out – 방법2

Branch checkout



# Q1. 윈도우 탐색기에서는 branch 어떻게 표현??

Branch checkout

- 특정 위치에 여러 개의 branch가 같이 존재한다.
  - C:\workspace\HelloWorld
  - 총 branch 3개가 존재 – branch1, branch2, branch3
- Eclipse에서 branch checkout하면 각 branch마다 파일이 다른 것이 보이는데, 윈도우 탐색기에서는 branch 마다 다른걸 어떻게 나타내지?

# A1. 윈도우 탐색기는 현재 branch를 보여줌

Branch checkout

- Eclipse에서 현재 checkout 되어 있는 branch를 보여준다.

## ➔ 확인해보자

- Eclipse와 윈도우 탐색기를 동시에 띄우고, eclipse에서 branch 변경시 윈도우 탐색기의 내용이 branch마다 변경되는 것 확인
- Branch1에서 branch1test.txt를 생성 후 branch1에만 commit, Branch2에서 branch1test.txt가 없는 것을 확인.  
윈도우 탐색기에서도 확인.

The background of the slide is composed of several overlapping geometric shapes. There are large orange shapes in the top-left and bottom-left corners, and a light blue shape in the top-right. A dark blue horizontal band runs across the middle of the slide, partially overlapping the orange shapes. The text 'To do list' is written in white on the left side of this dark blue band.

## To do list

# 아직 해보지 않은 것

---

- Revert
- Eclipse를 사용하여 Github에 있는 repository clone하기
- GUI tool을 활용해서 clone, branc생성, 삭제, checkout, commit, push, pull 등 해보기
  - Sourcetree app 활용해서 해보기

# 과제

---

- **Sourcetree app을 활용하여 Git 기본 사용 방법 동영상 작성**
  - <https://www.sourcetreeapp.com/>
  - branch 생성, 삭제, checkout, commit, clone, push, pull 진행
  - 해당 과정을 동영상으로 제작하여 본인 git 설명 기존 홈페이지에 링크 공유
- **제출 방법**
  - Github 해당 repository 홈페이지 url 제출
    - 예) <https://{본인id}.github.io/{repository명}>
  - 정보시스템개요/과제제출/과제2\_Sourcetree활용.xls 에 url 기재
- **제출 기한**
  - 4월 10일 금요일 오후 6시



**Thank you**