DESIGN VERSIONING GUIDE

Contents

GIT

01-1

Git Install

깃 설치

Learn More GIt : https://git-scm.com/

Download Address

https://git-scm.com/

Git Document

https://git-scm.com/book/ko/v2

Git Install Check

Terminal > Enter “git”

* Normal Screen, if it is not, install again

01-2

Git Basic

깃 기본

Learn More GIt : https://git-scm.com/

1. Git Install
2. Git Basic

Repository [= repo]

프로젝트의 디렉토리나 저장 공간. 당신의 컴퓨터 안의 로컬 폴더가 될 수도 있고,
깃허브나 다른 온라인 호스트의 저장 공간이 될 수도 있다.

Fork vs Merge Request

포크(Fork)는 저장소에 있는 UI Kit이나 다른 도메인의 디자인 파일을 그대로 복제하는 기능이다.
복제된 파일은 원본에 영향을 주지 않고 자유롭게 프로젝트를 진행 할 수 있다.
작업이 끝난 파일을 원본 파일 반영하기 위해서는 병합 요청(Merge Reuqest)해야 한다.

*3-1 참조

Commit

커밋(Commit)하면, 진행되는 프로젝트 시점의 저장소 상태를 기록해 프로젝트를 이전으로 돌아가거나
복원할 수 있는 체크포인트를 가질 수 있다.

*4-4 참조

Pull & Push

풀(Pull)은 다운로드(Download), 푸시(Push)는 업로드(Upload)로 이해하면 쉽다.
커밋(Commit)을 통해 작업된 기록물들을 원격저장소인 깃랩(Gitlab)에 올리기 위해서 푸시(Push)를 한다.
미리 업로드한 작업물을 내려받거나 다른 작업자가 작업한 작업물을 내려 받기 위해서는 풀(Pull)을 하면 된다.

*4-5,6, 7 참조

Branch

여러 명이 하나의 프로젝트에서 작업할 때, 파일을 받아서 이름을 변경하고 자신만의 버전을 만들어 작업한다.
일반적으로 브런치(Branch)는 이럴때 미리 자신 만의 브런치(Branch)를 만들 수 있는 기능이다.
작업을 끝낸 후, 프로젝트의 메인 디렉토리인 “마스터(Master)”에 브랜치를 다시 “병합(Merge)”한다.

*4-8 참조

GITLAB

02-1

GitLab Infomation

모든 프로젝트, 소스들의 버전 관리 저장소

Address

gitlab.coupang.net

Login Account

Coupang Groupware Account

GitLab UI Repository

gitlab.coupang.net/release/coupang-design-convention

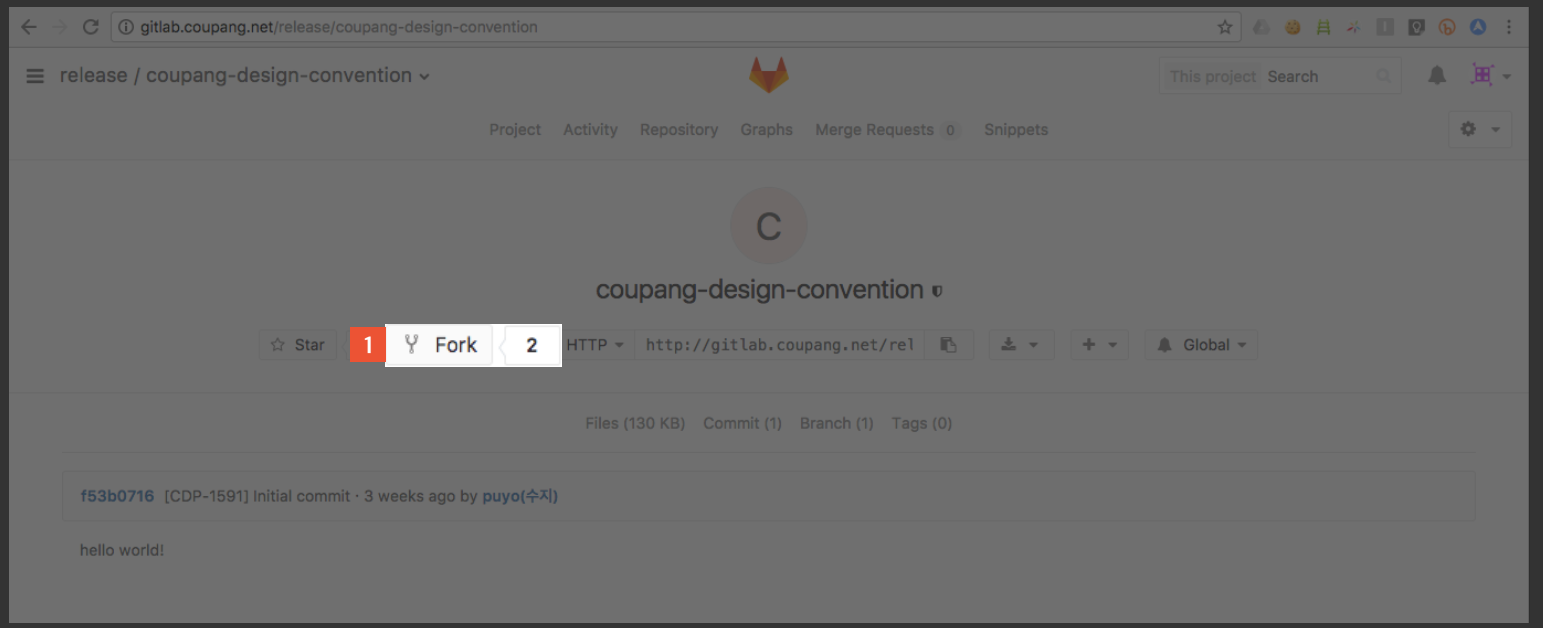
GITLAB TEAM

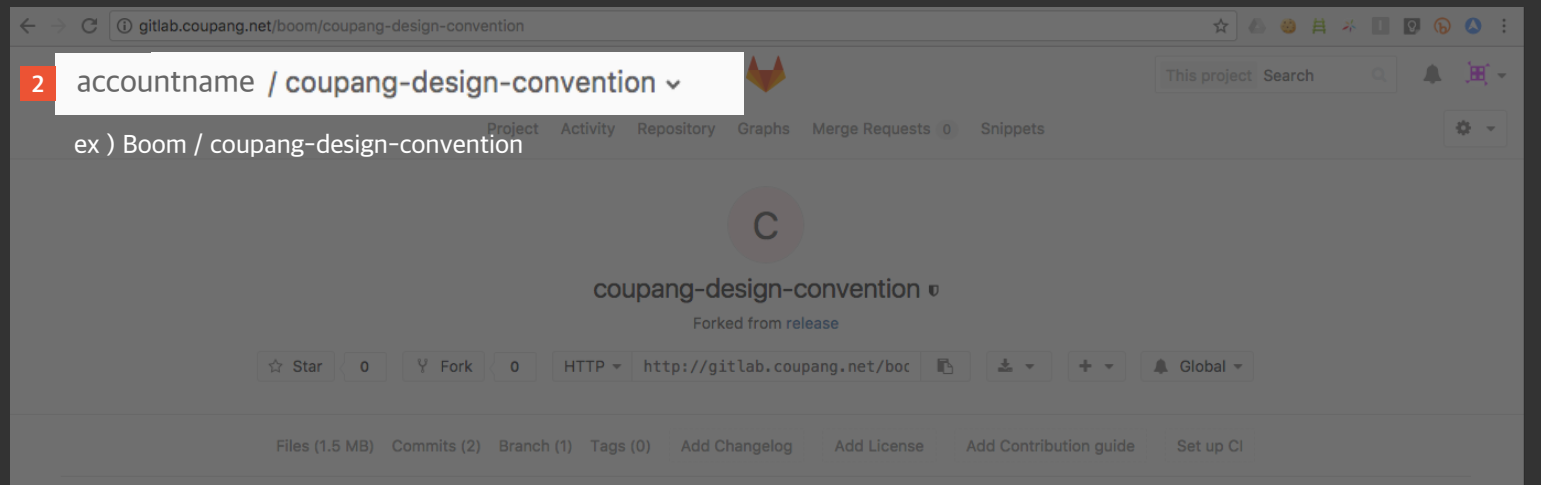
03-1 Fork Parent Repository

Team Setting

1. Fork
2. Add Members
3. Merge Request

Fork GitLab UI Repository > Check Address

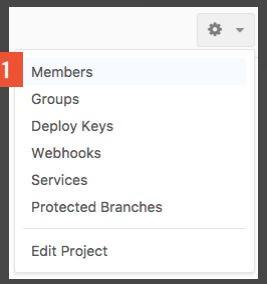


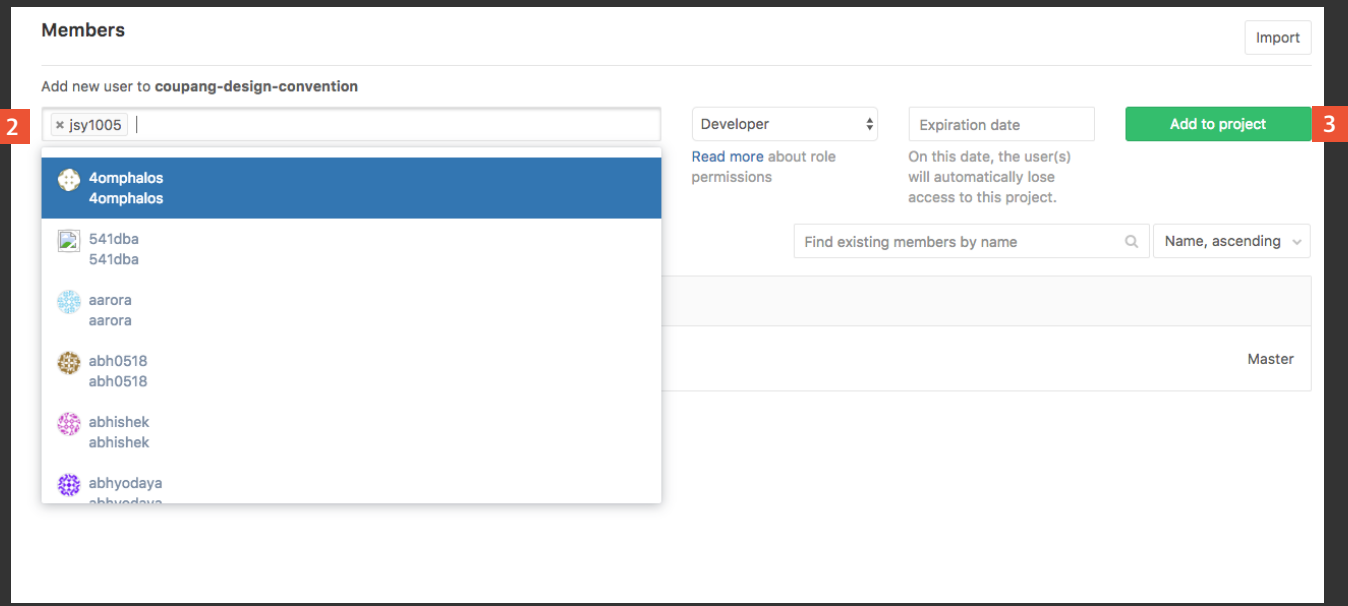


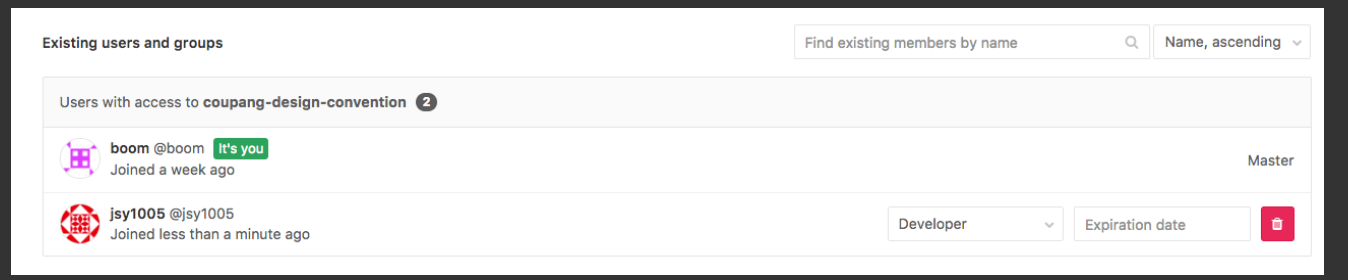
03-2 Add Members

1. Fork
2. Add Members
3. Merge Request

Repository Setting : Members > Search Members > Role Setting > Add to project







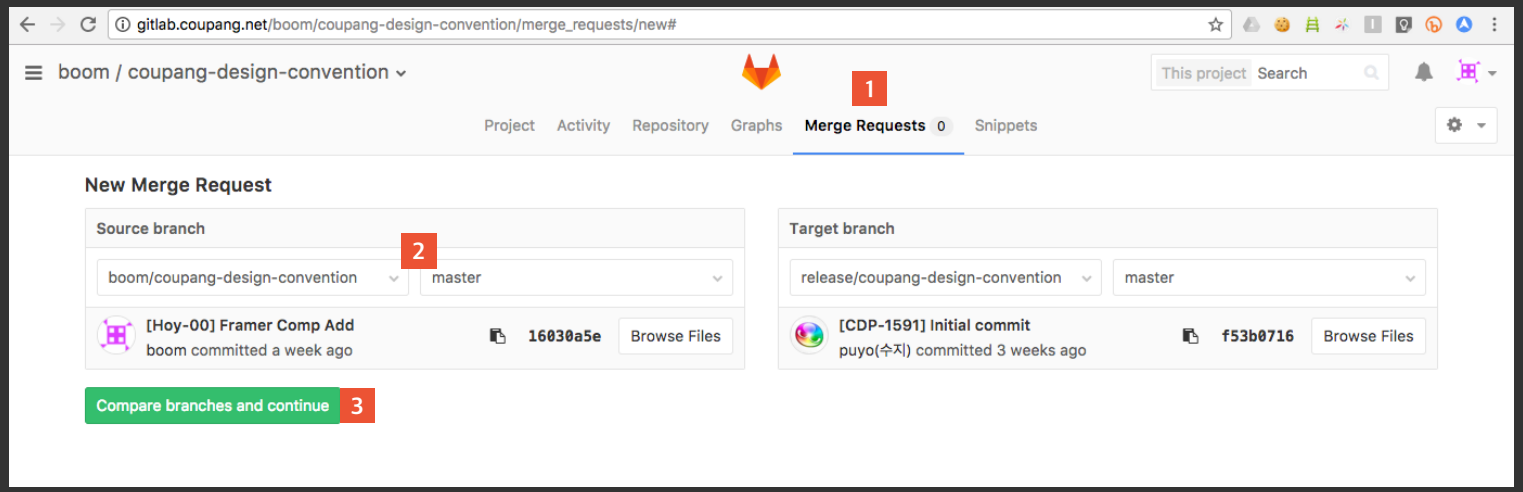
03-3

Merge Request after Project

1. Fork
2. Add Members
3. Merge Request

   (Project End or Source Update)

Merge Requests (Original Repository) > Select Branch > Compare branches and continue



GIT FOR DESIGNER

04-1

Software Intstall

깃 관리 프로그램 설치

1. Software Install (Desktop)
2. SourceTree Setting
3. Clone Repository
4. Commit
5. Push & Pull
6. Branch
7. Merge
8. Checkout

Source Tree (Free)

www.sourcetreeapp.com

Folio (Fee : $49)

http://folioformac.com/

Only Github Software

Github (Free)

https://desktop.github.com/

Kactus (Fee)

https://desktop.github.com/

ETC Design Version Control Software

Abstract(Fee)

www.goabstract.com

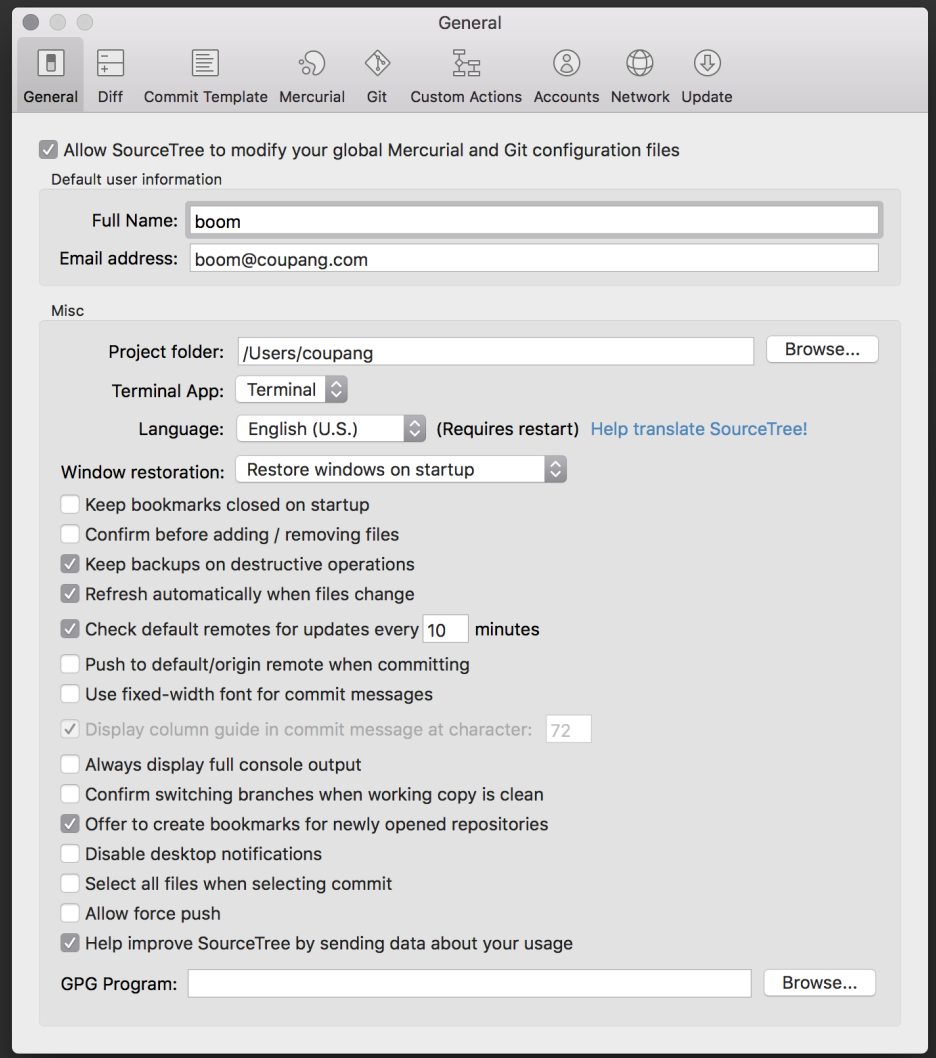
04-2

SourceTree Setting for Git

깃 관리 프로그램 설치

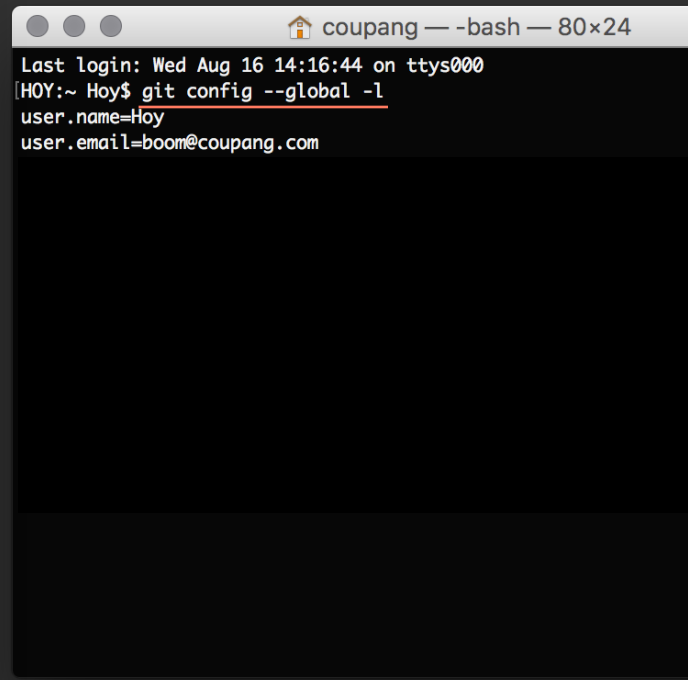
1. Software Install (Desktop)
2. SourceTree Setting
3. Clone Repository
4. Commit
5. Push & Pull
6. Branch
7. Merge
8. Checkout

SourceTree > Preference > General : Enter User.Name / User.Email



Check Terminal

"git config - - global -l"

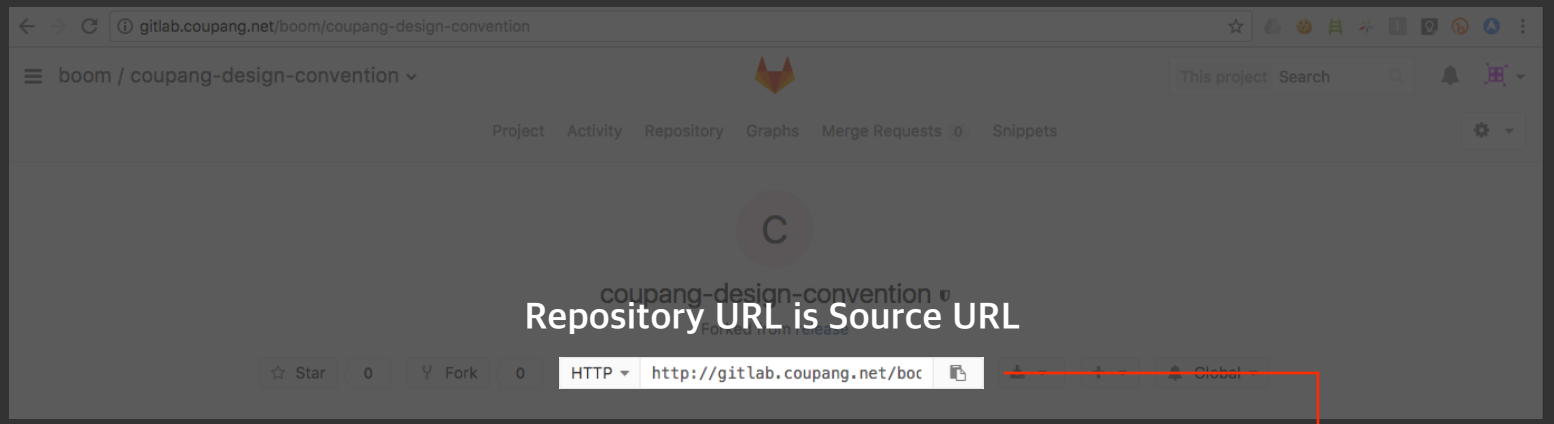


04-3

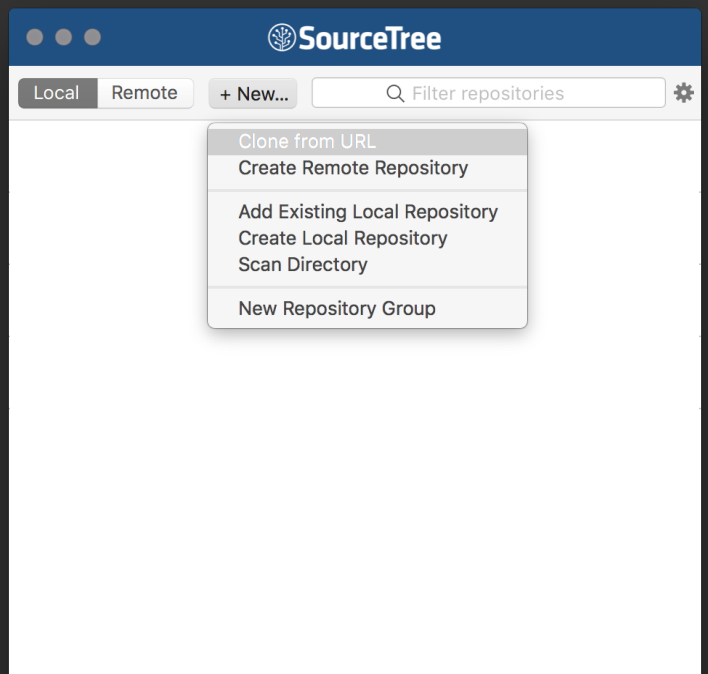
Clone Repository

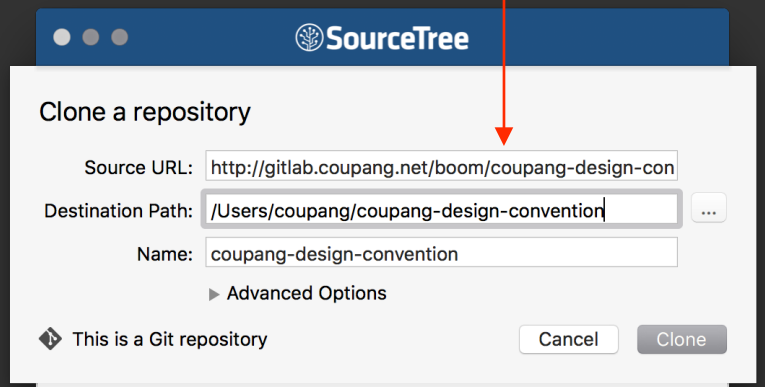
원격 저장소 복제

New > Clone from URL > Copy & Paste Repository URL > Local Path Setting > Clone



Copy & Paste





Destination Path = Local Path

Name = Folder Name(Please Dont change) = Repository Name

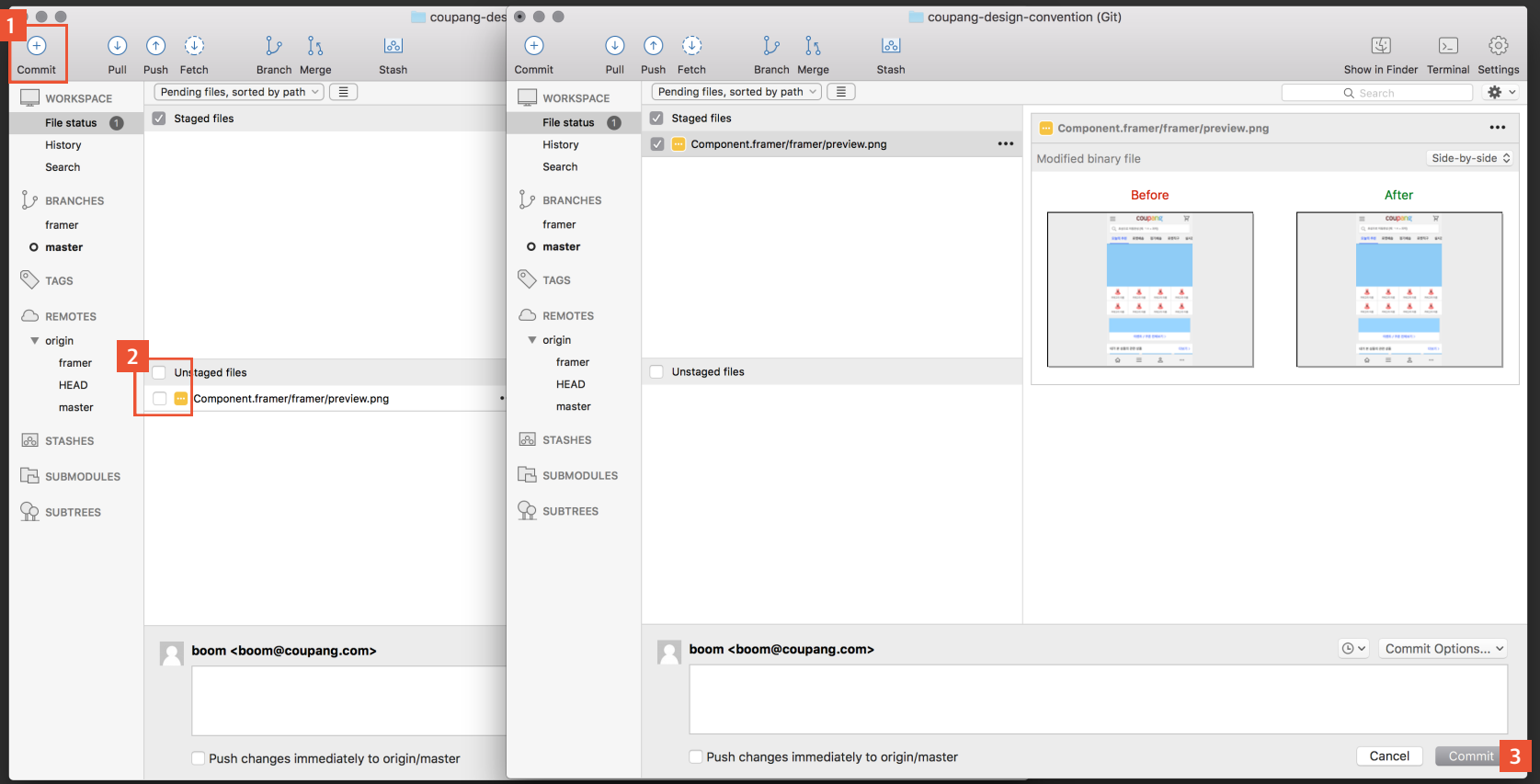
04-4

Commit after end of project

프로젝트 끝난 후 Git 관리를 위한 커밋

Commit > Check Unstaged files(or Final) to stage > Write Commit Message > Commit

1. Software Install (Desktop)
2. SourceTree Setting
3. Clone Repository
4. Commit
5. Push & Pull
6. Branch
7. Merge
8. Checkout



Commit Messege Rule

If don't follow this rule, Could not push Anything after commit.

[Nickname-00] Domain-ProjectName-Status(Update, Issues)

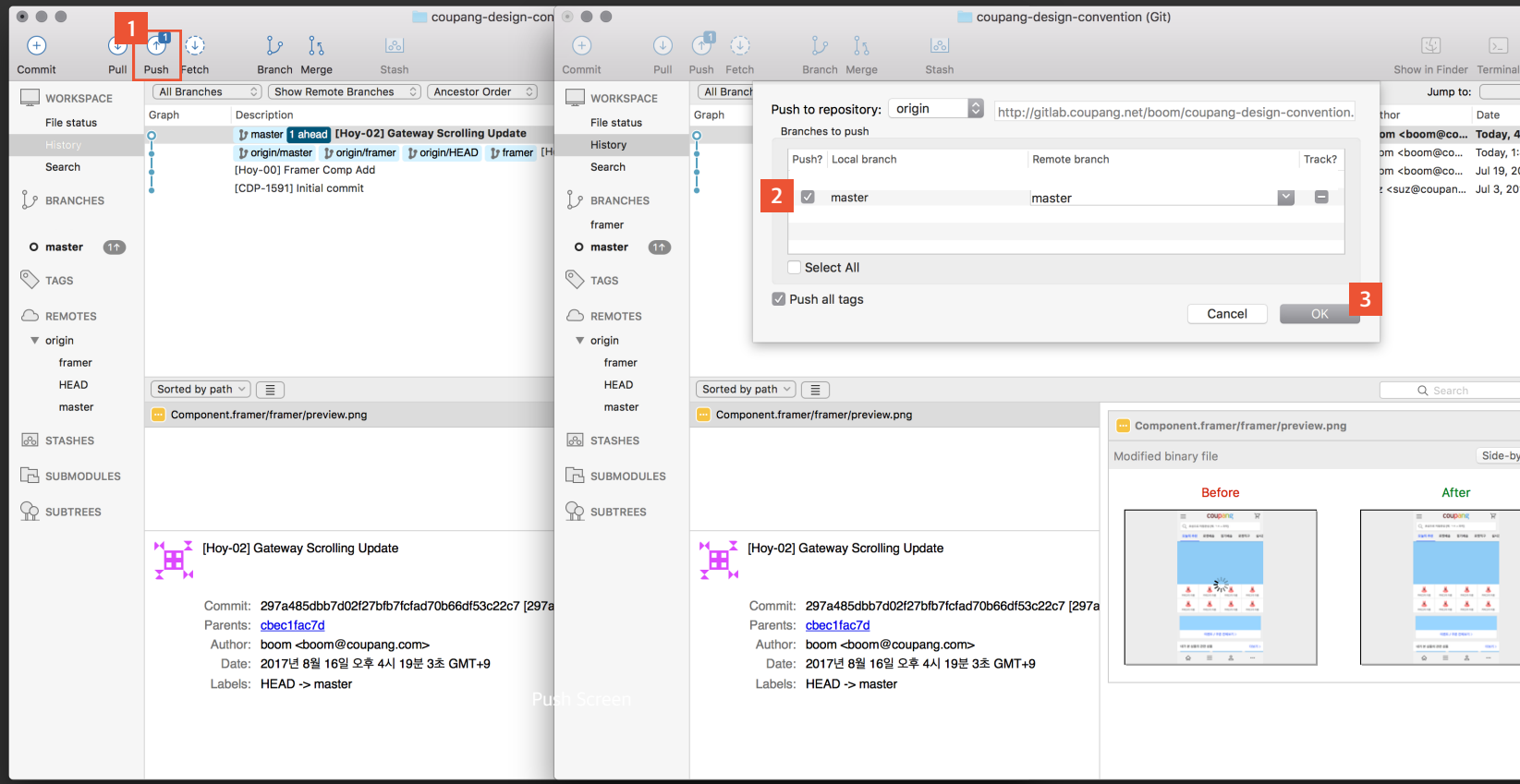
ex) [Hoy-00] Gateway-ThemeShop-Category-Add

04-5

Push to Repository

커밋 후 저장소 업로드(Push)

Commit > Push > Select Branch > Push



Commit Screen

Push Rule

If don't follow this rule, Could not push Anything.

1. Do Commit first
2. If there is(are) Pull, Have to Pull first before Push.

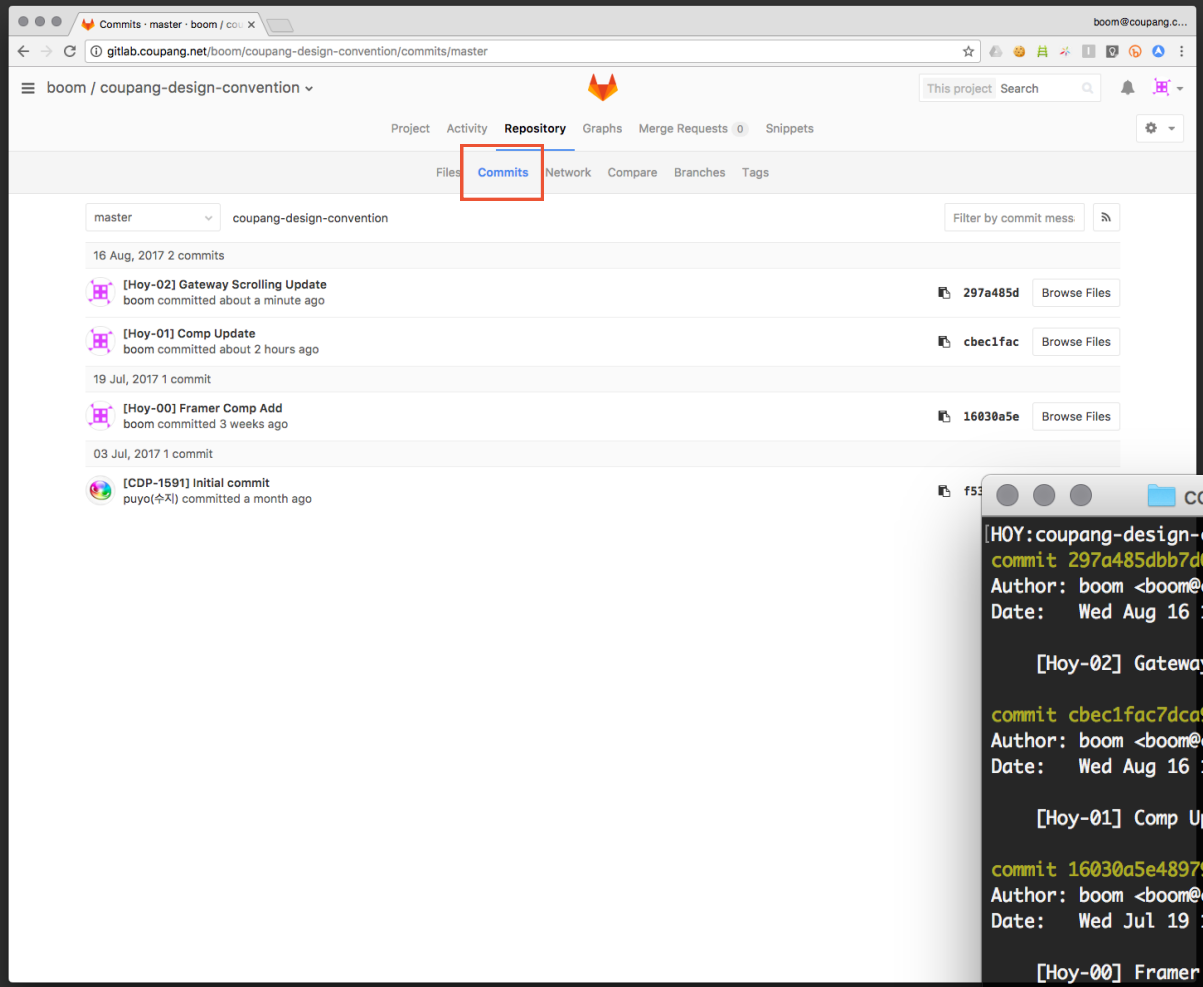
04-6

Check Push & Commit History

푸시 및 커밋 히스토리 확인

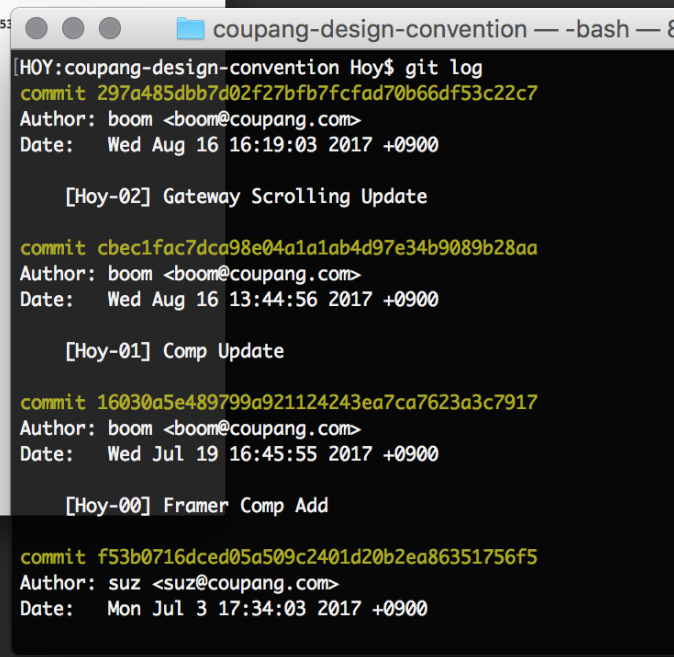
1. Software Install (Desktop)
2. SourceTree Setting
3. Clone Repository
4. Commit
5. Push & Pull
6. Branch
7. Merge
8. Checkout

Gitlab : Repository > Commits



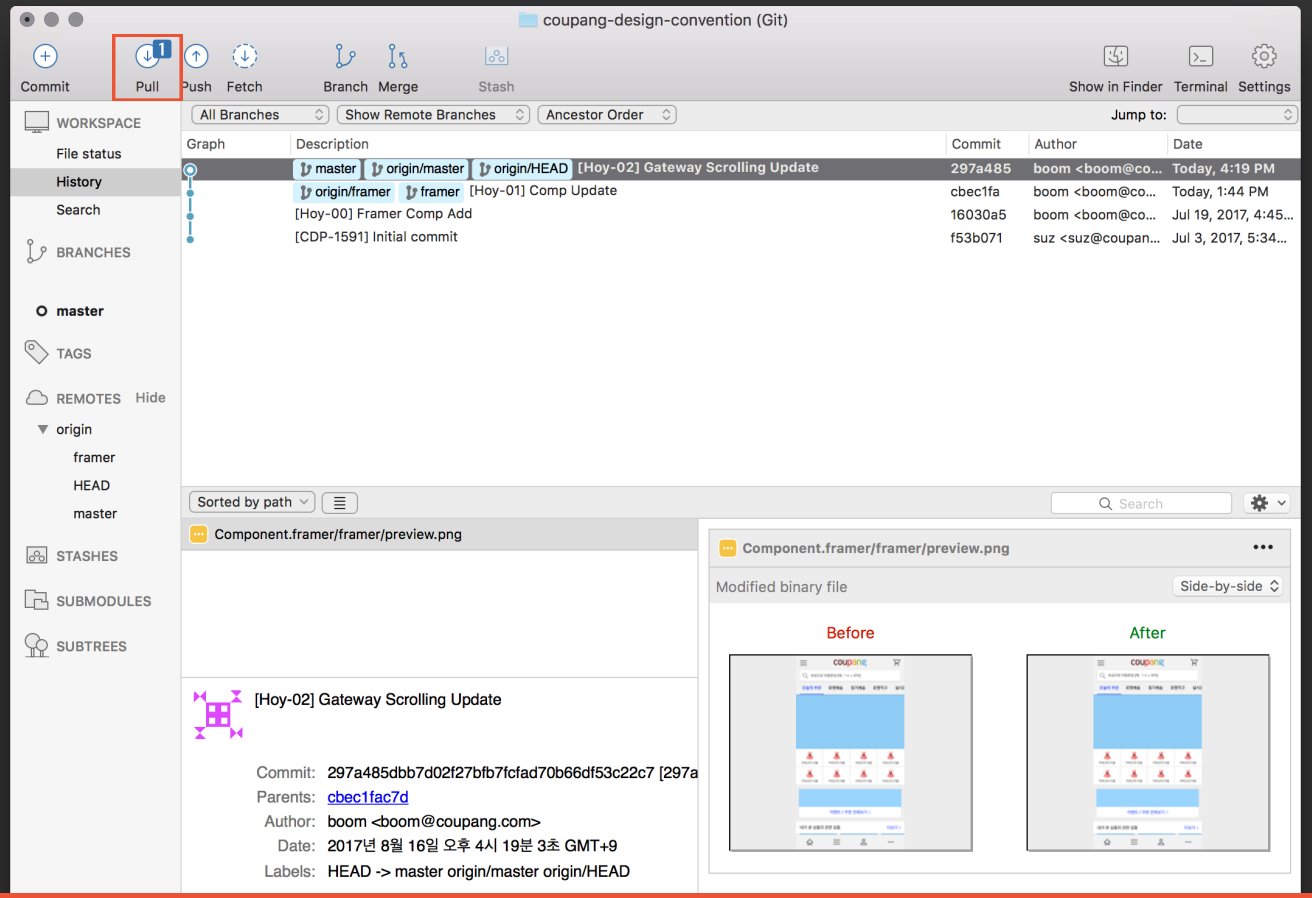
Check Terminal

"git log"



04-7 Pull

파일 당겨오기 : 해당 폴더(Branch)의 파일이 업데이트 됩니다.



Pull Rule

Pull first before the Push

1. If project is worked in common file(ex: UI Kit), Have to Branch first for your design files.

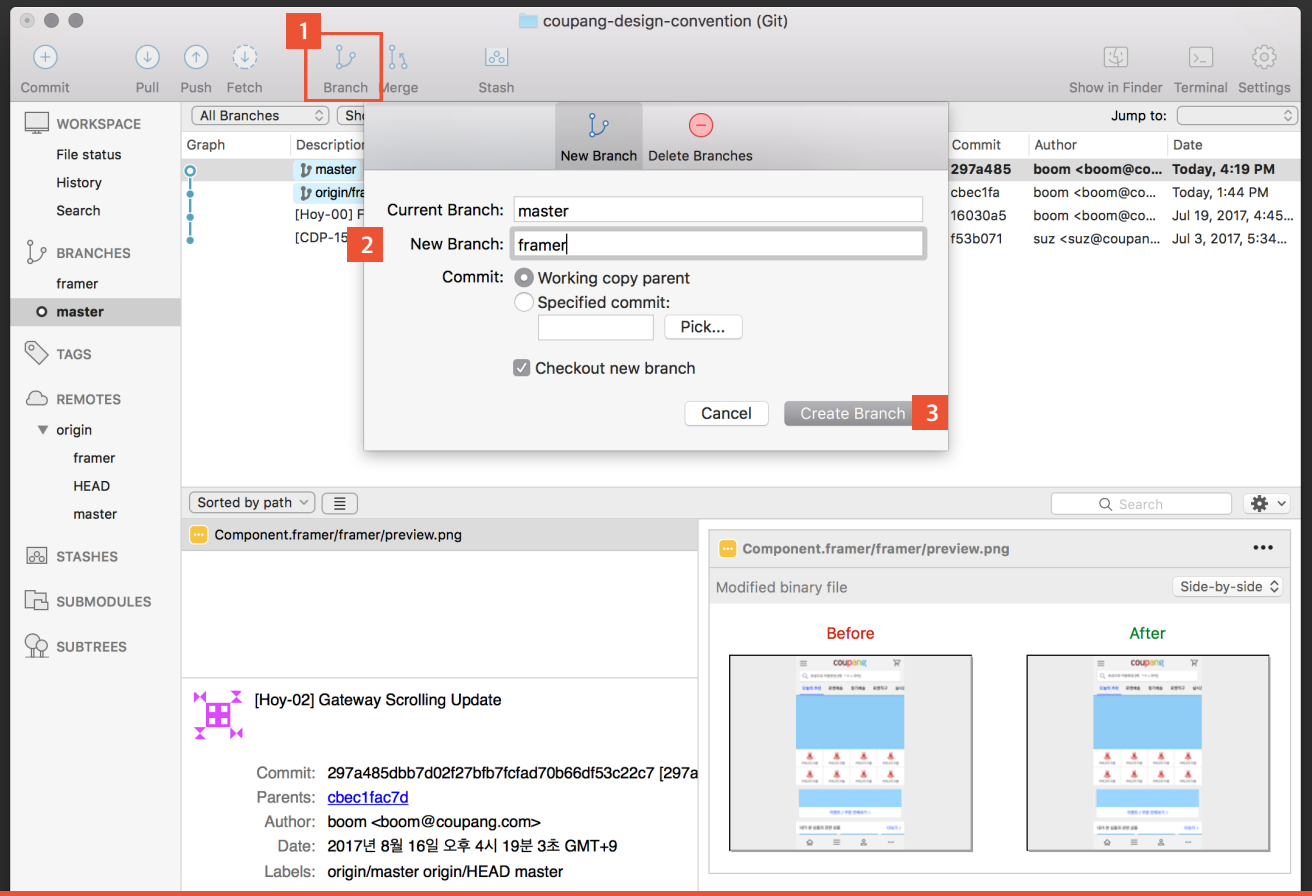
If you have already branched, Just Pull.

04-8

Branch

자신만의 버전 만들기

Branch > Input Branch Name (Nick or Feature) > Create Branch



Branch Rule

Purpose : Safe of lossing design files

Designer

1. Each Designer, Each Branch
2. Each Feature, Each Branch

Manager

1. Each Domain, Each Branch
2. Each Project, Each Branch

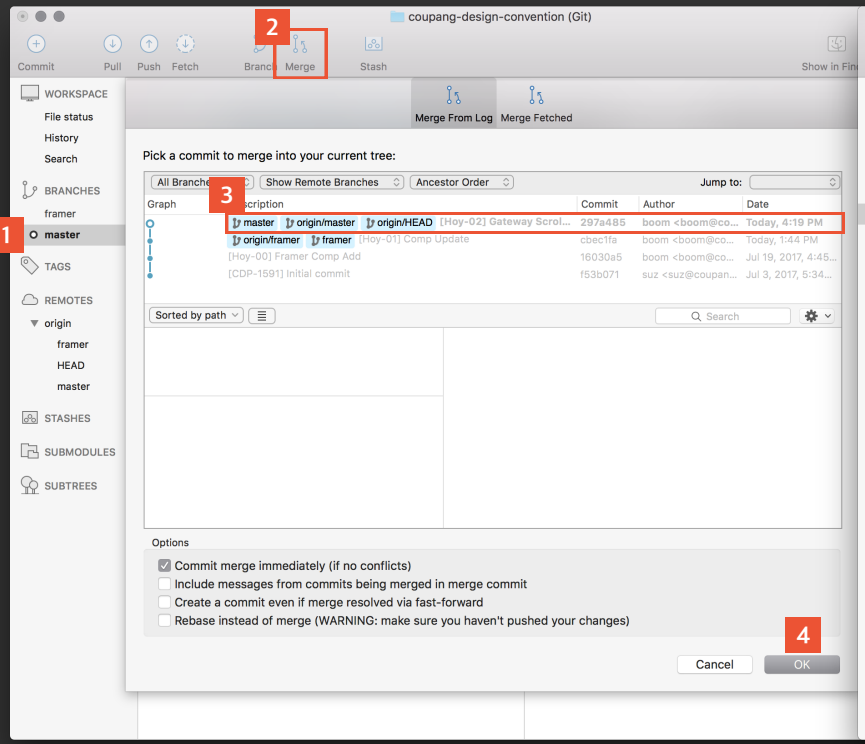
04-9

Branch File Merge

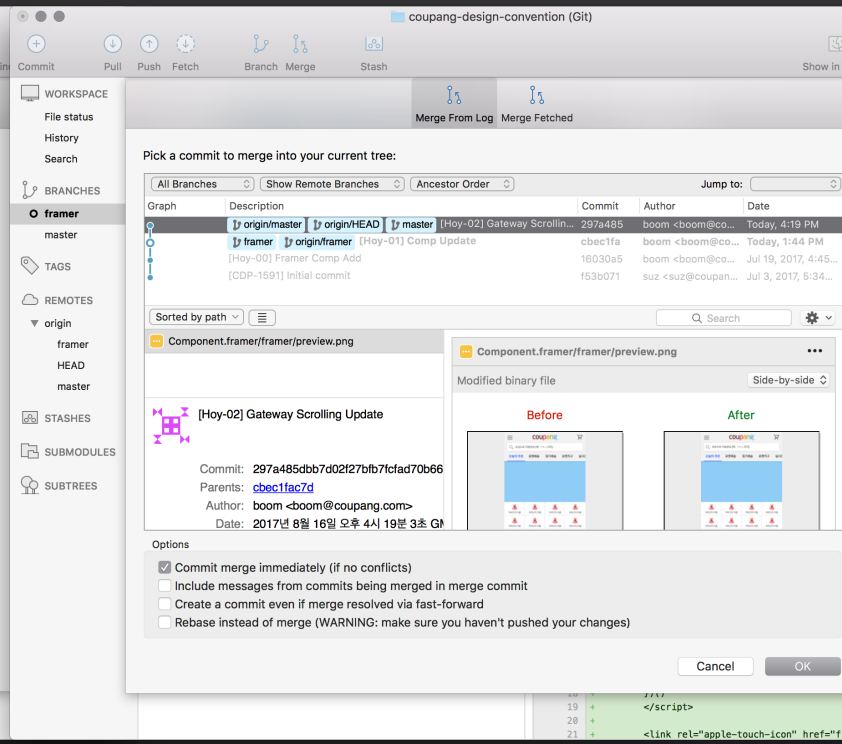
브런치 파일 병합하기

Select Unchanged Branch > Merge > Select The Lastest Commit > Ok

1. Software Install (Desktop)
2. SourceTree Setting
3. Clone Repository
4. Commit
5. Push & Pull
6. Branch
7. Merge
8. Checkout



Merge to master from branch



Merge to branch from master

Merge Rule

If you choose the wrong branch, files might be disappeared.
but Choose woking branch is showed Working Design Files

1. Merge after Commit
2. Merge have to base from changed branch
3. When common ui is pulled in Master Branch, Base is Working Branch

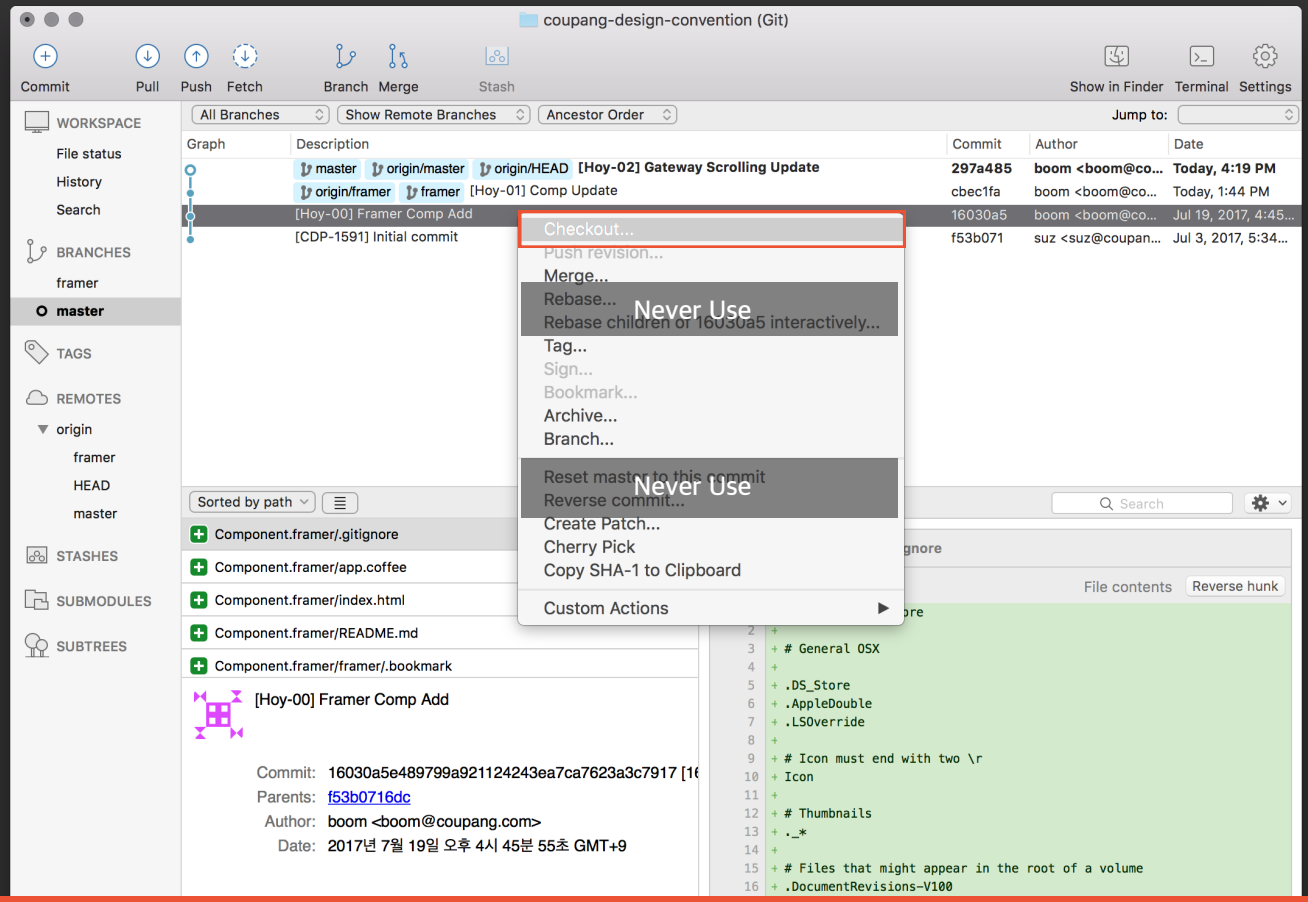
04-10

Checkout

과거 작업물 확인

1. Software Install (Desktop)
2. SourceTree Setting
3. Clone Repository
4. Commit
5. Push & Pull
6. Branch
7. Merge
8. Checkout

Mouse Right > CheckOut > Check Folder(the file is changed to this commit)



Checkout Rule

If you choose the wrong branch, files might be disappeared

1. if you change after selected commit, make branch first.

Please, Use only what you know