



Pull Request(PR) Guide

Git Desktop

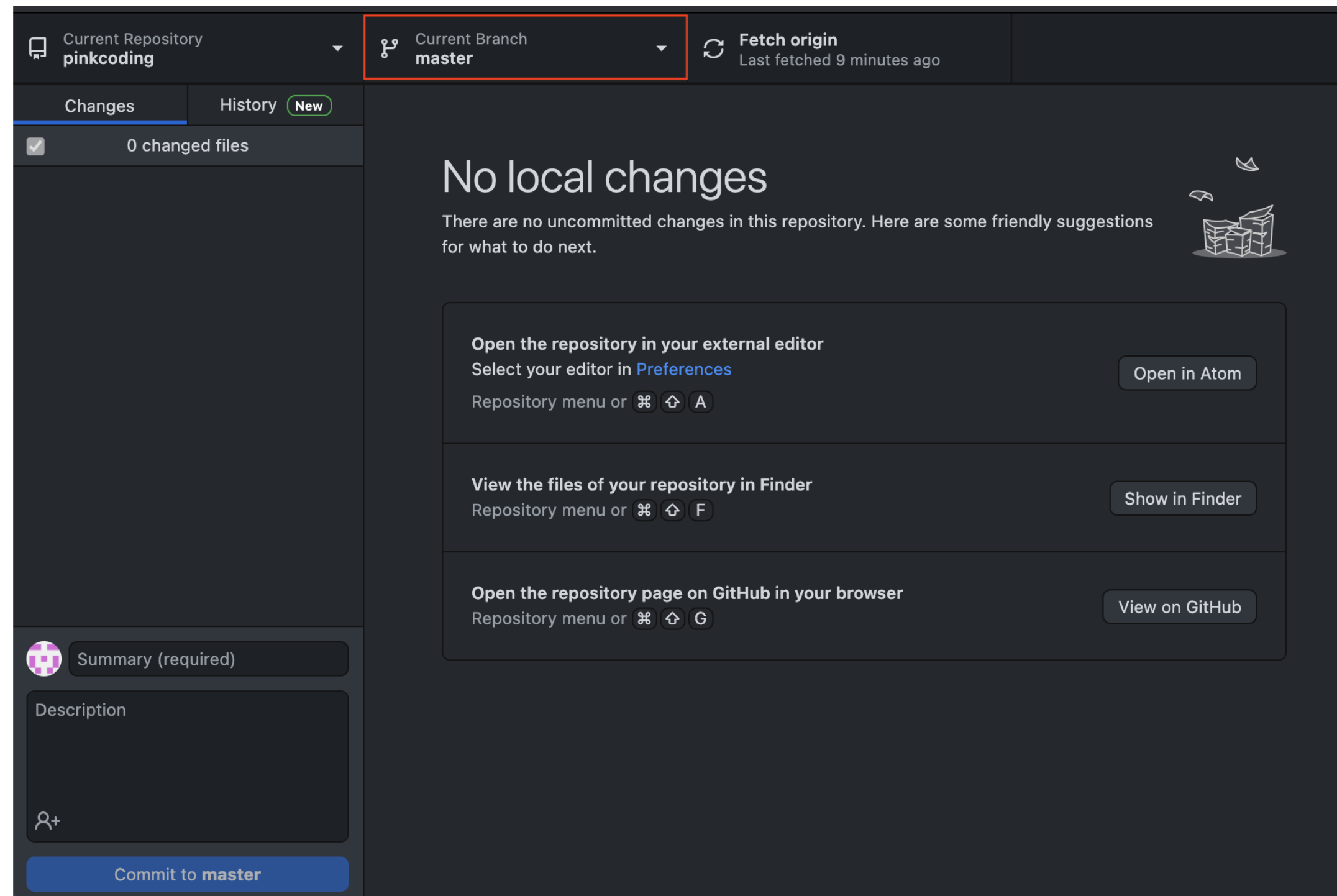


```
const interbyorg = study.lead_organization === interbyorg ? true : false;
const status = filterByStatus ? study.status === filterByStatus : true;
const matchStatus = filterByStatus ? status : true;

function filterStudies({ studies, filterByOrg, filterByStatus }) {
  return studies.filter(study => {
    return study.lead_organization === filterByOrg &&
    (filterByStatus ? study.status === filterByStatus : true) &&
    (filterByStatus ? status : true);
  });
}
```

Git Branch 생성하기

with Git Desktop



Git desktop에서 기본 branch가 master 인 경우

<시나리오>

A프로그램



새로운 기능

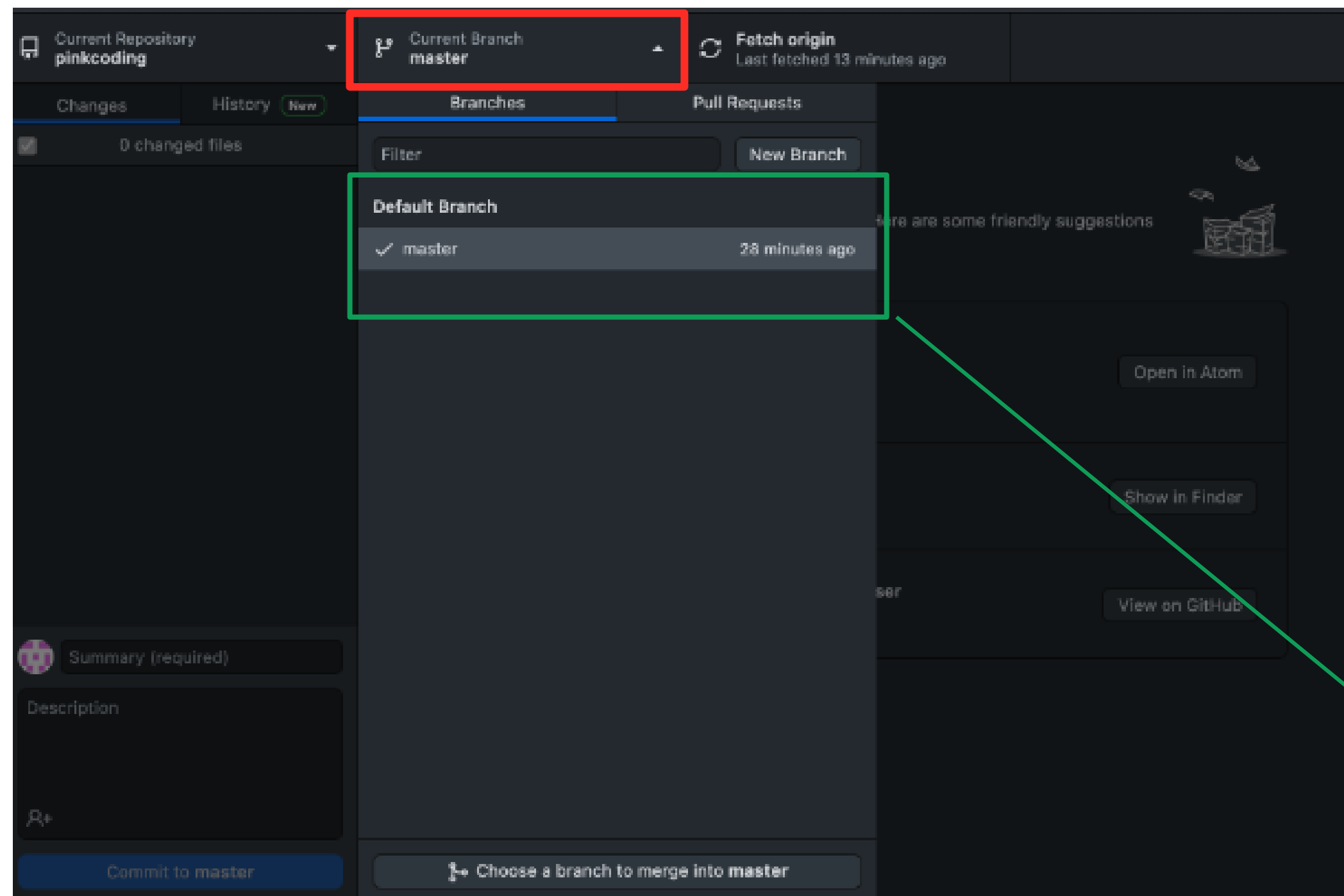
A프로그램에 새로운 기능을 추가하려 합니다.
그러나 바로 만들어지지 않습니다.

그래서 현재 A 프로그램 프로젝트를 하나 본떠서
이곳에 새로운 기능을 추가하고 테스트 해봅니다.



Git Branch 생성하기

with Git Desktop



Current Branch master 버튼을 누른 후 모습

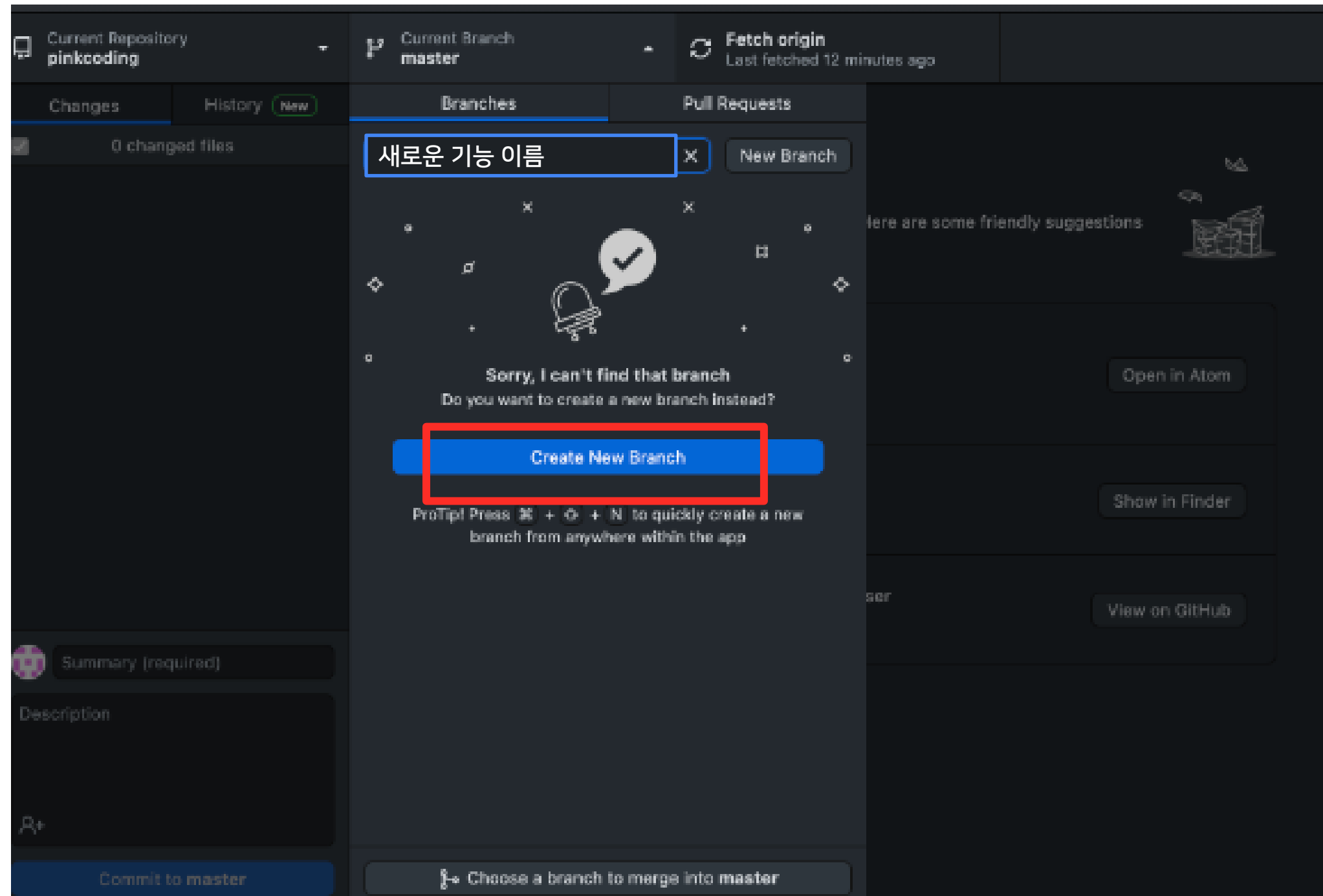
Current Branch
master

현재 Branch가
master 라는 의미

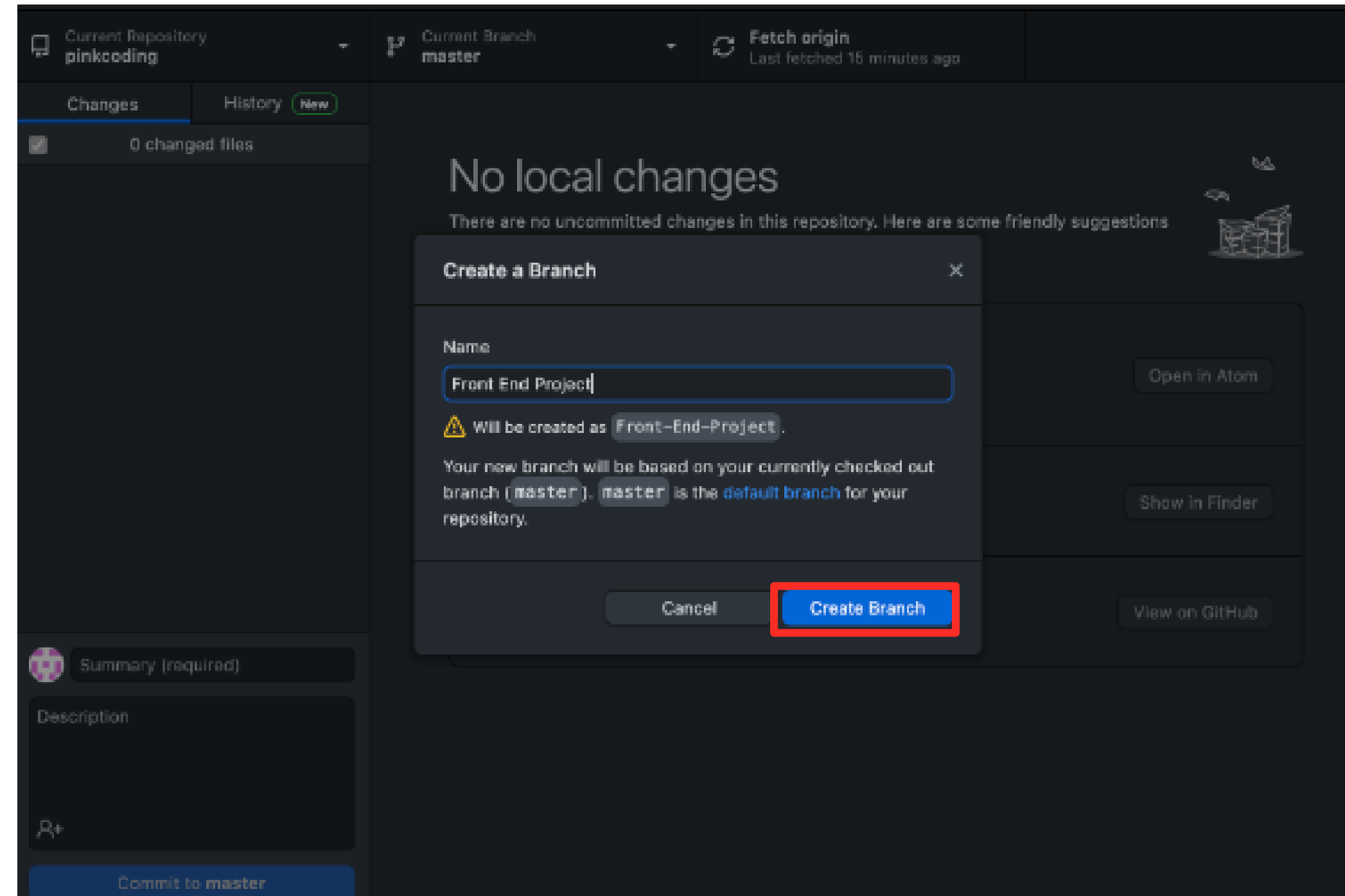
Default Branch = master OR main

Git Branch 생성하기

with Git Desktop



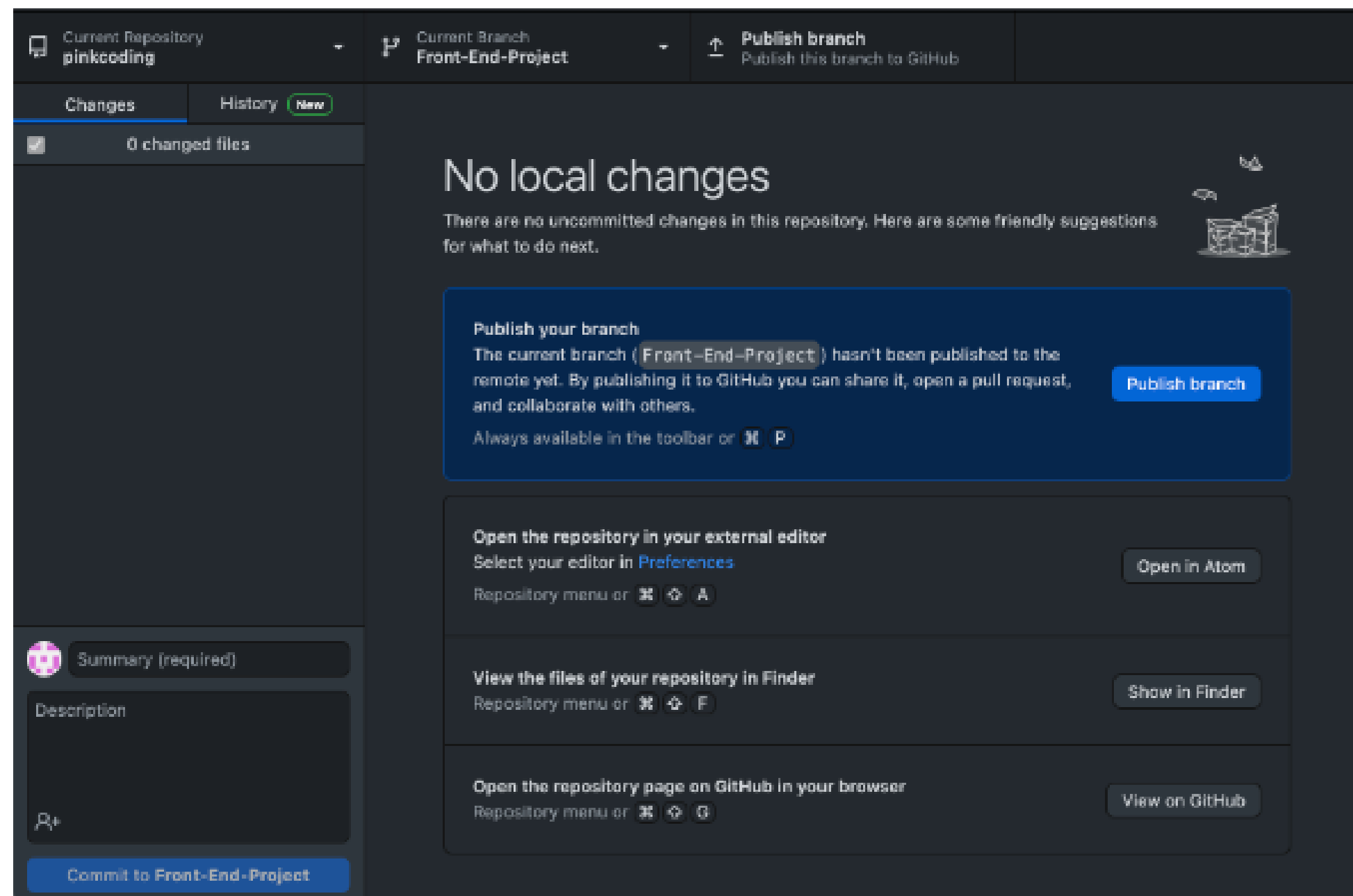
Create New Branch 클릭



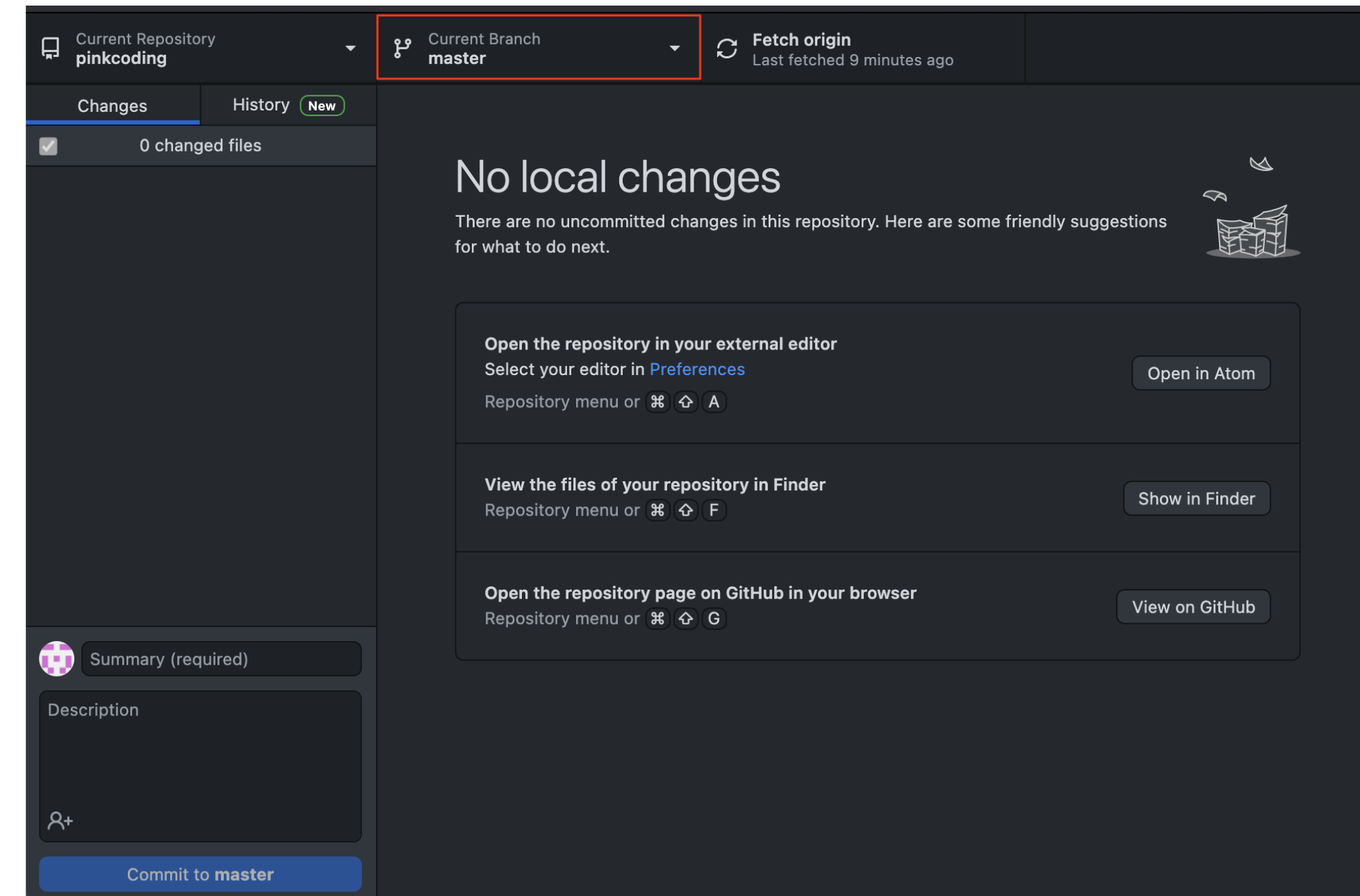
Create Branch 클릭

Git Branch 생성하기

with Git Desktop



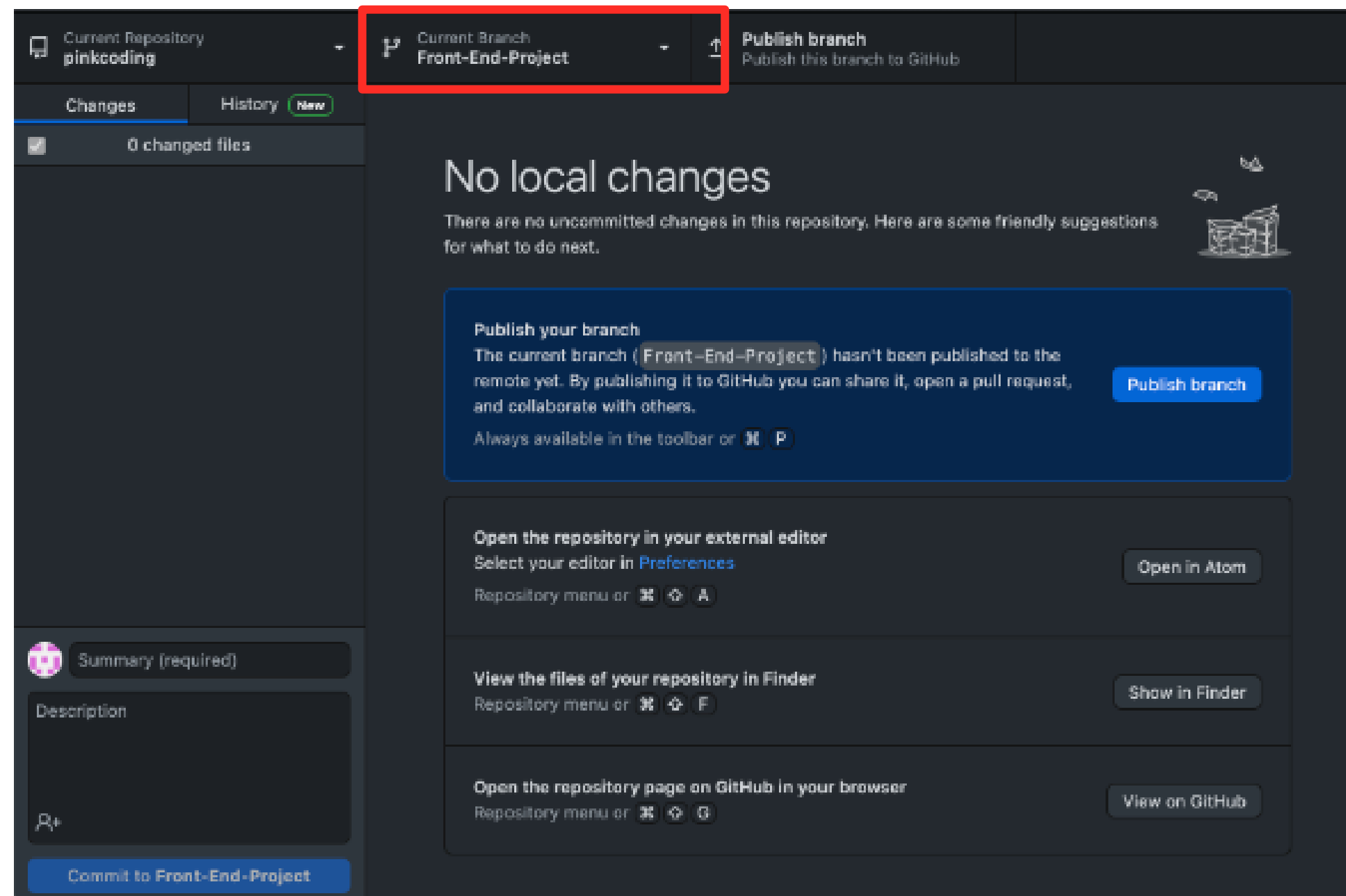
새로운 Branch를 생성 후 모습



새로운 Branch를 생성하기 전 모습

Git Branch 생성하기

with Git Desktop



더 이상 master branch가 아니다!

깨알 Tip!

Branch 이름은 공백을 허용하지 않아서 공백이 '-'로 변경됩니다.

ex) Front End Project

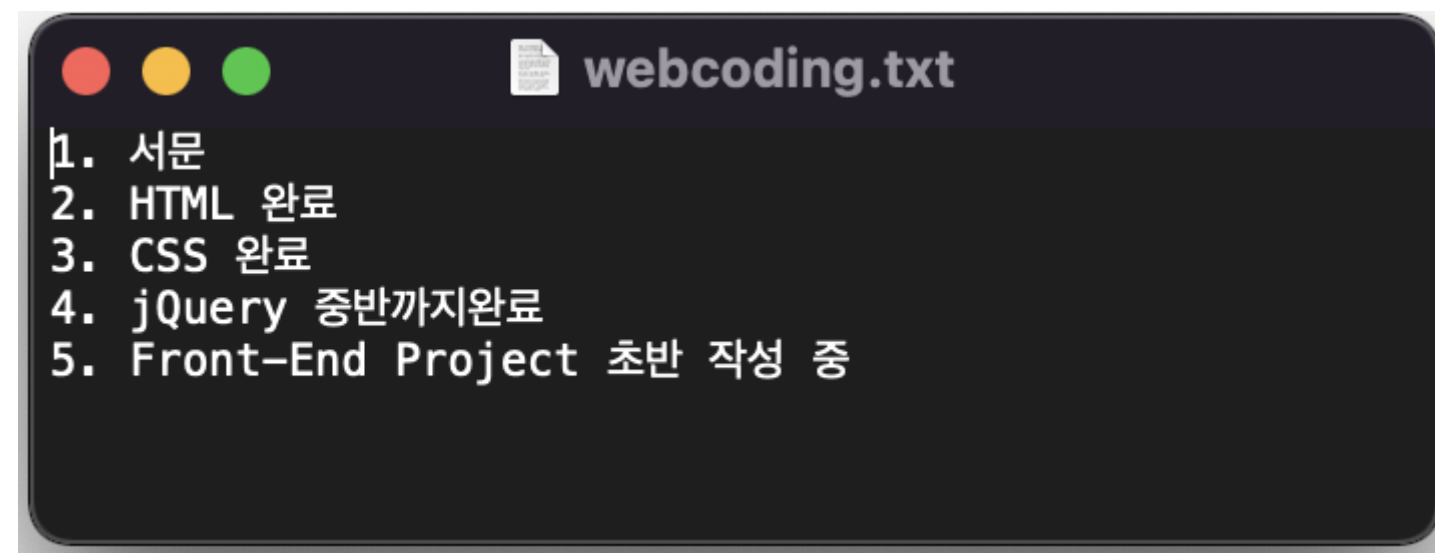


Front-End-Project

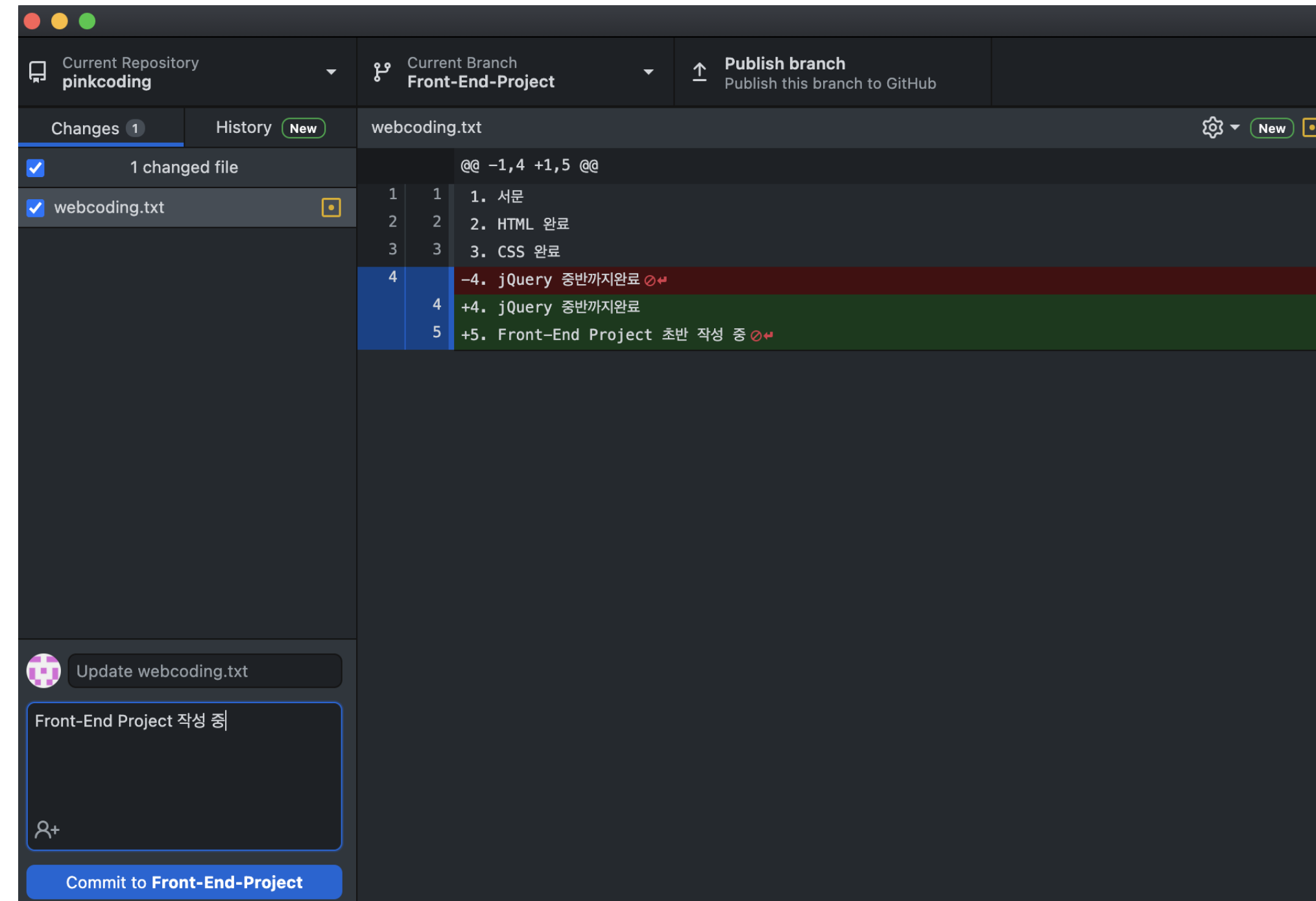


Git Branch 생성하기

with Git Desktop



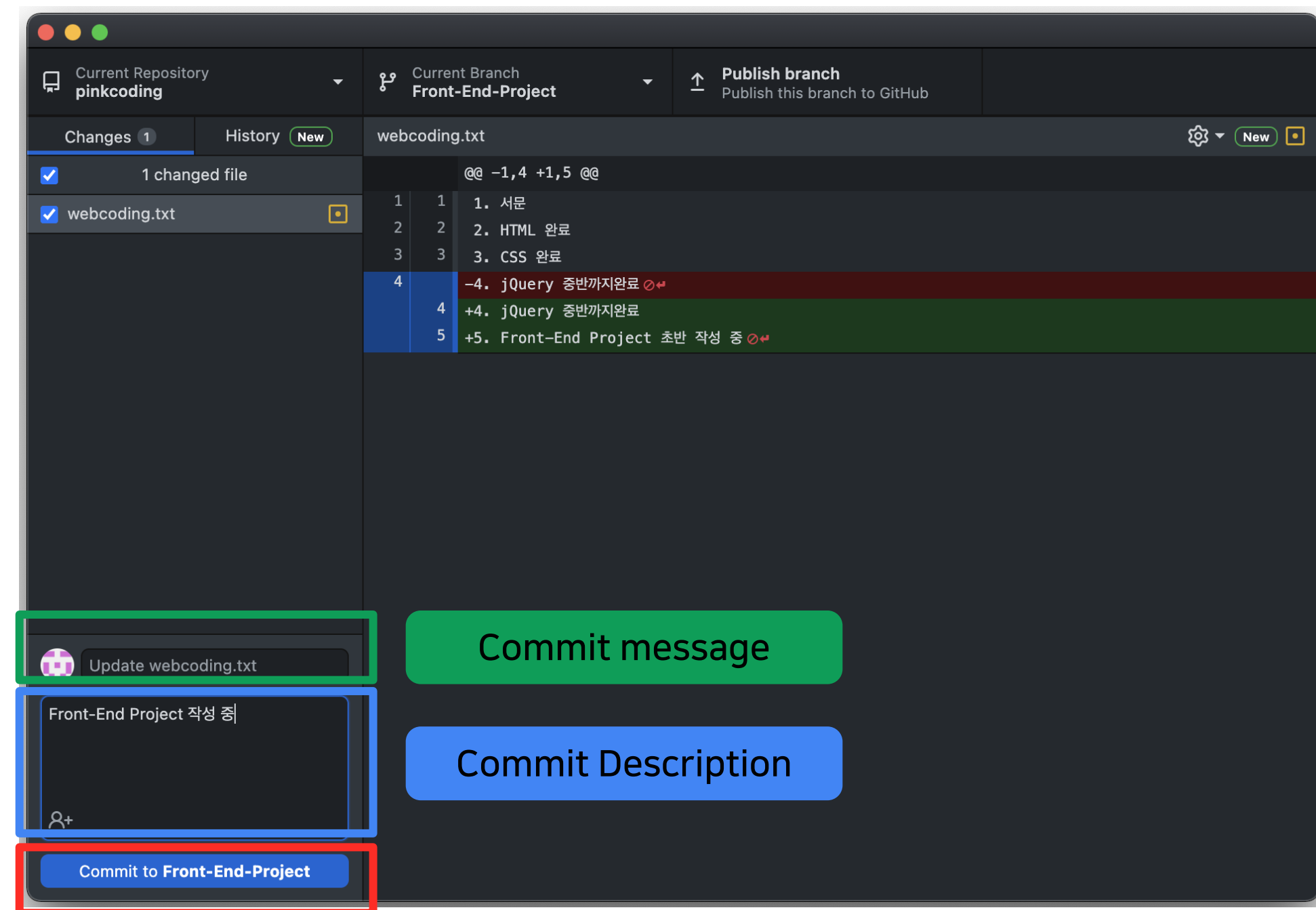
새로운 Branch에서 작업한 txt파일



현재 Branch는 Front-End-Project이므로 변경된 사항은 Front-End-Project에서만 적용됩니다.

Git Branch 생성하기

with Git Desktop



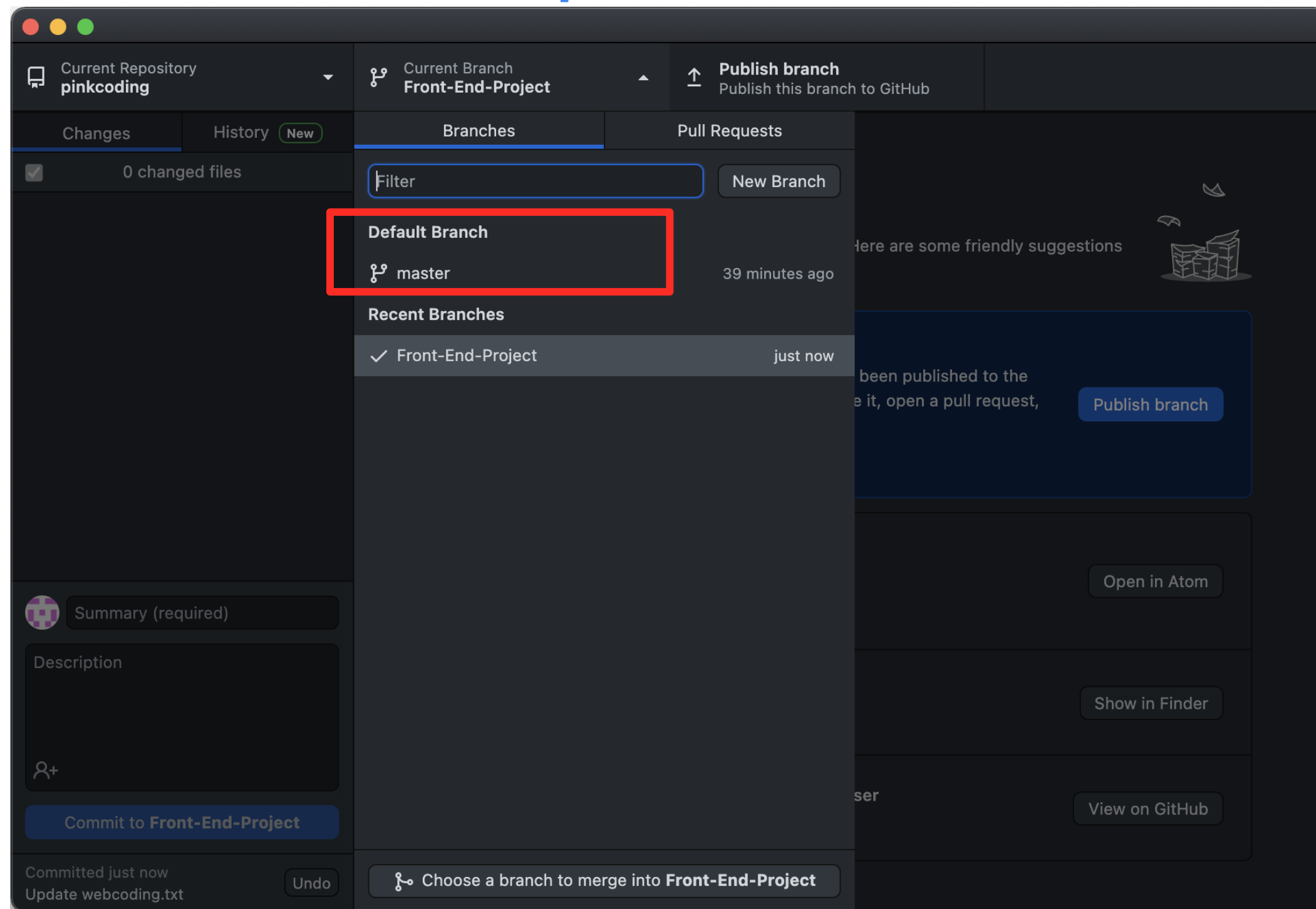
Commit to Front-End-Project를 클릭

깨알 Tip!

Commit message
Commit description
두 가지는 **별도의 규칙**이 있으니
참고하자!

Git Branch 생성하기

with Git Desktop



다시 Branch를 master로 변경하기

Front-End-Project



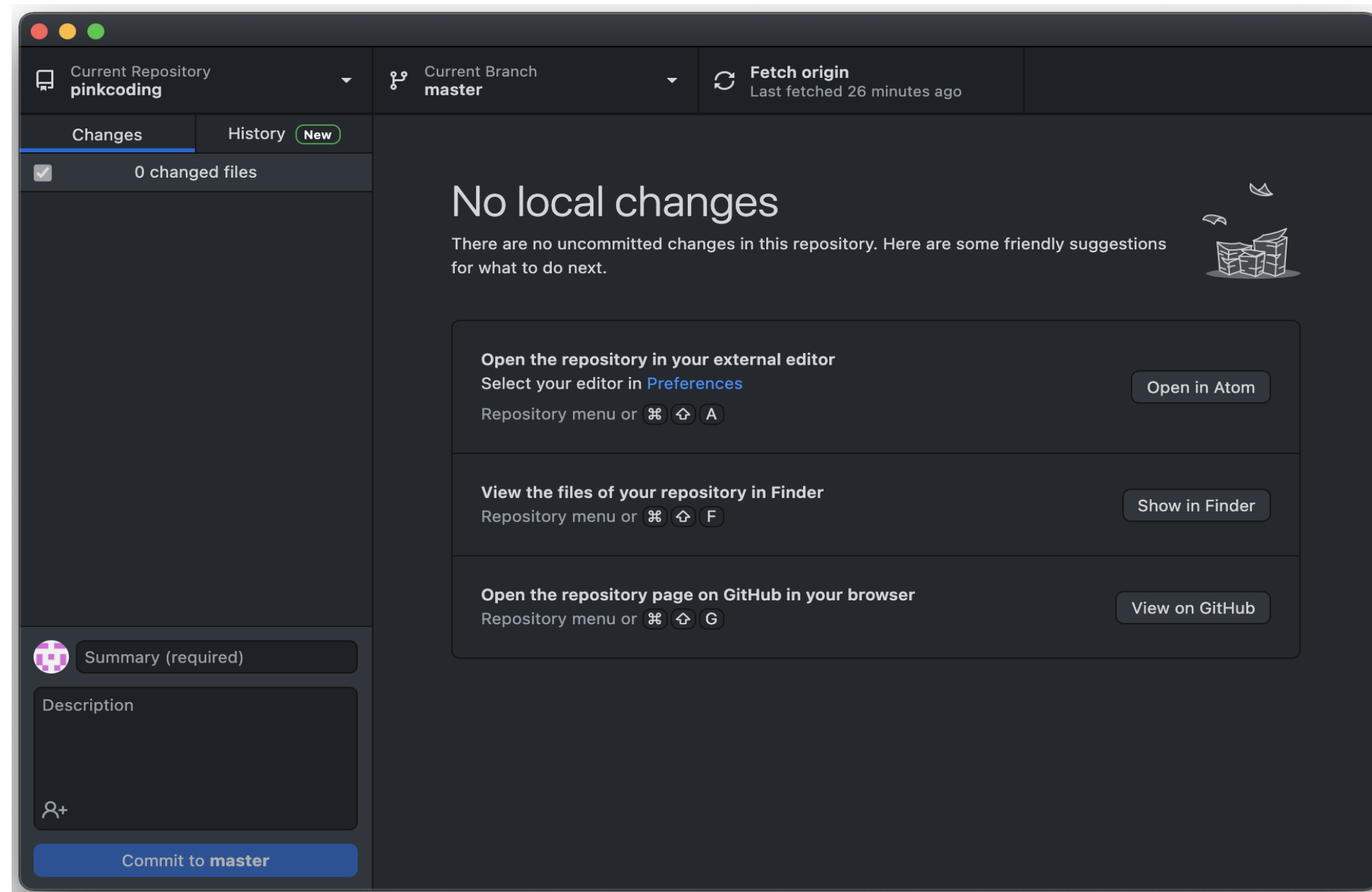
master

브랜치를 변경하는 것을 checkout(체크아웃)이라고 부른다.



Git Branch 생성하기

with Git Desktop



다시 master로 돌아옴

Front_End_Project에서 했던 작업은 없어졌다!

WHY

master branch에 돌아온 순간 master에서 했던 작업이 나타나기 때문임.

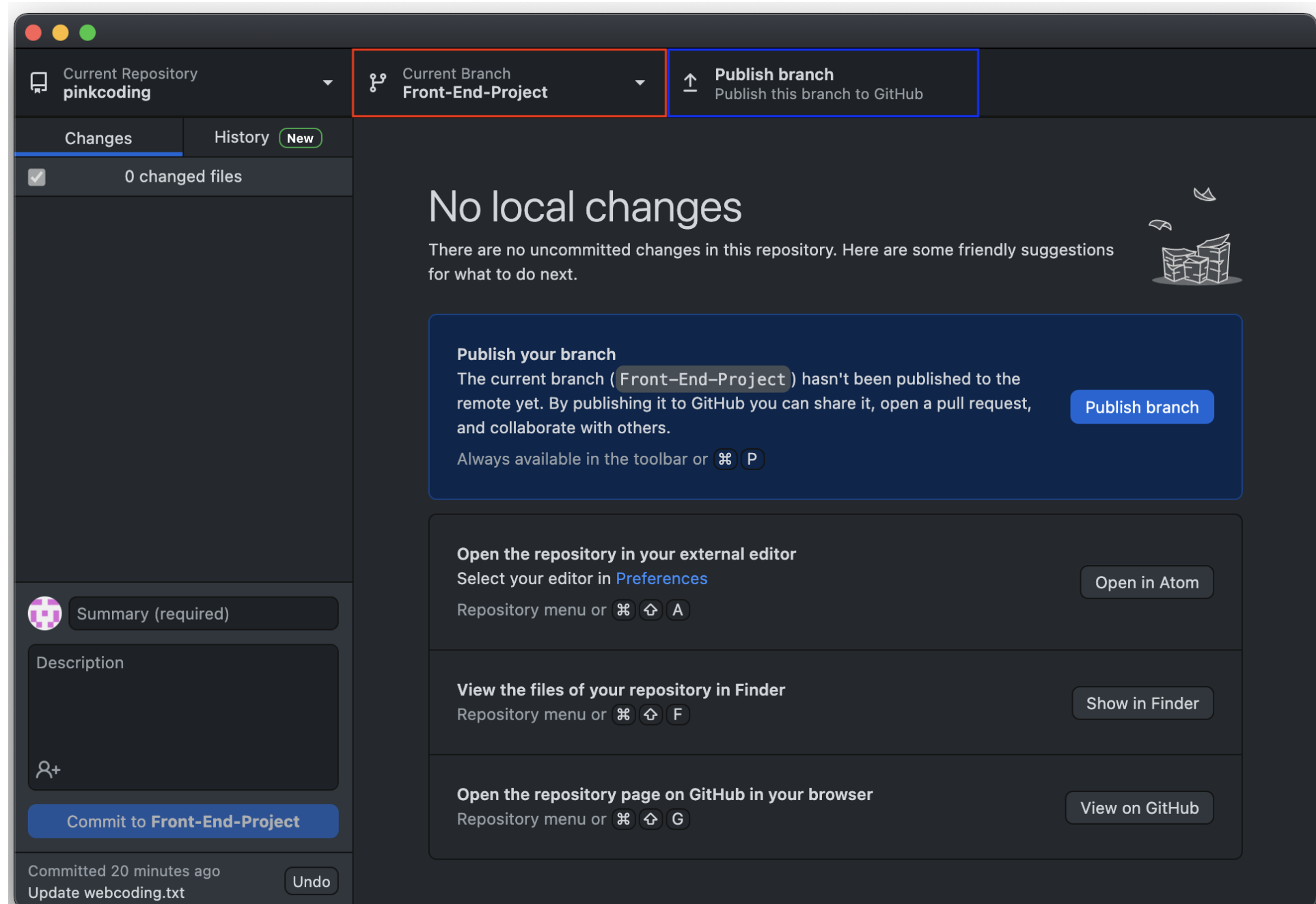
이렇게 브랜치를 생성해서 다른 큰 작업은 다른 브랜치에서 작업을 할 수 있습니다.

이제 Front-End-Project branch를 master branch에 합쳐봅시다.



Git Branch 합치기

with Git Desktop



Front-End-Project로 checkout 후
publish branch 선택

Front-End-Project

로컬 저장소(내컴)



Front-End-Project

원격 저장소(Git)

Front-End-Project에서 작업한 내용을 master에서도 보기 위해

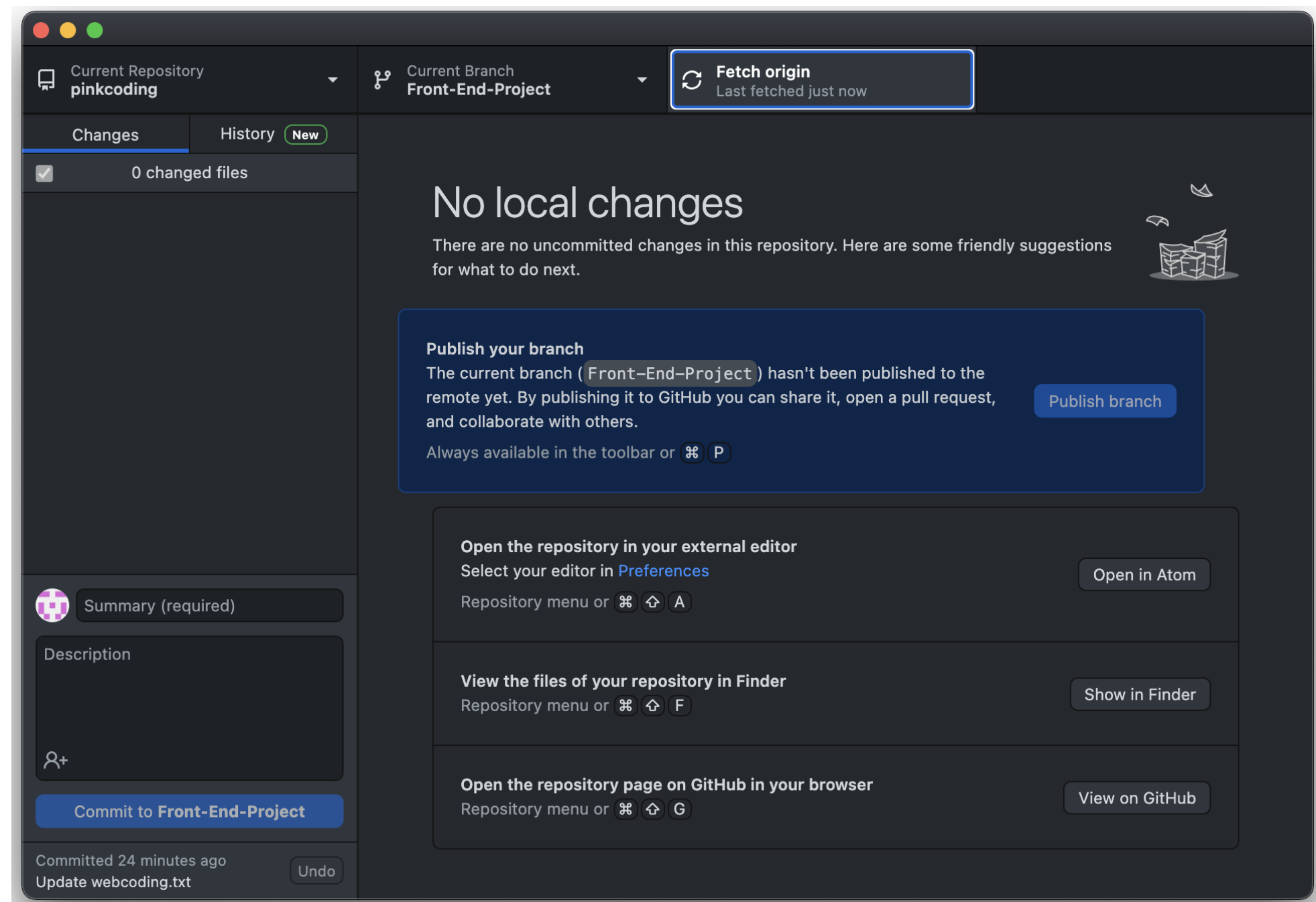
로컬 저장소에서 원격 저장소로 올려야 합니다.

이때 **publish branch**를 클릭합니다.



Git Branch 합치기

with Git Desktop



Fetch origin 클릭

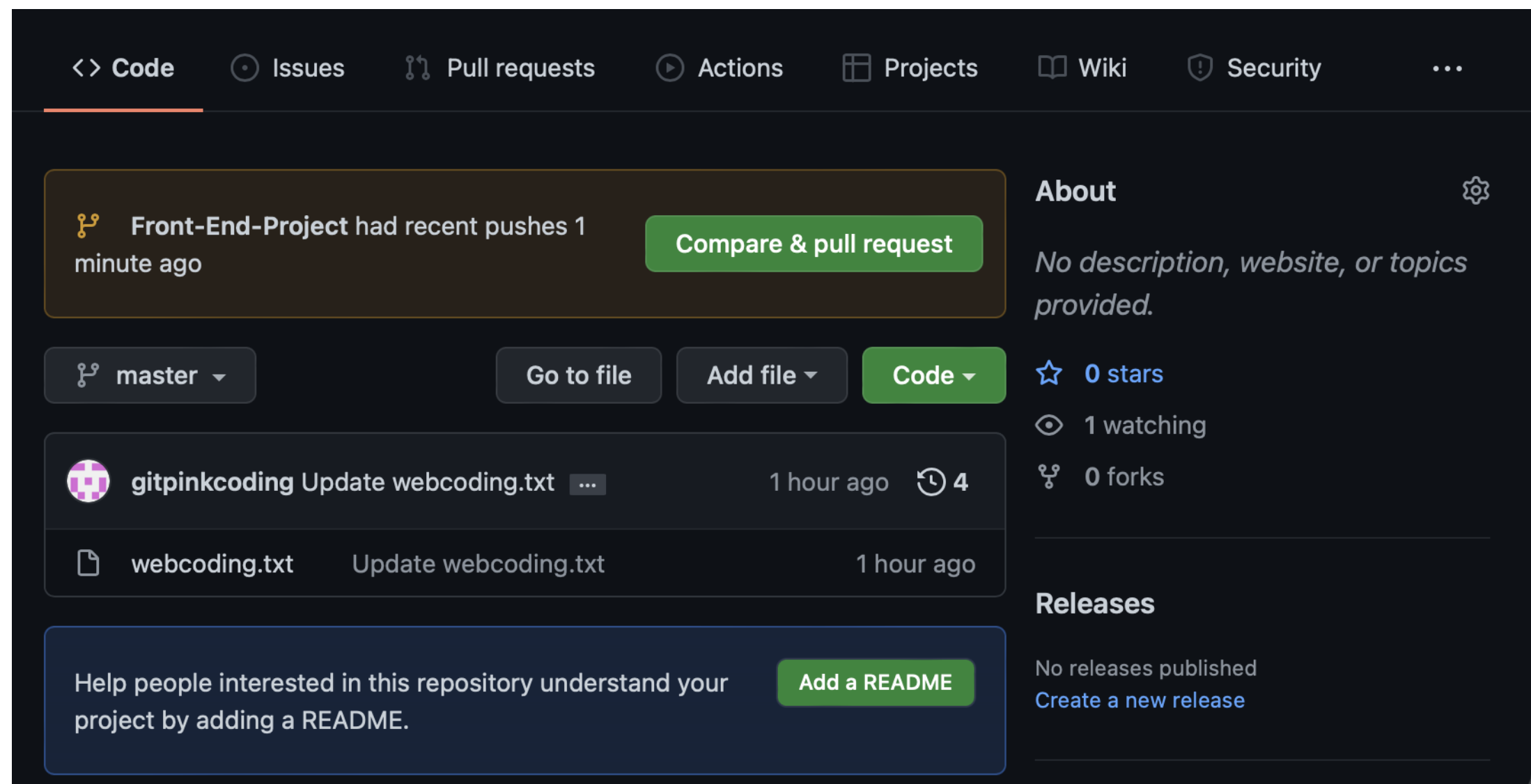
Fetch???

fetch는 **원격 저장소의 커밋들을 로컬 저장소로 가져온다.**
자동으로 merge(병합)하지 않으니 직접 확인 후 merge하는 과정이 필요



Git Branch 합치기

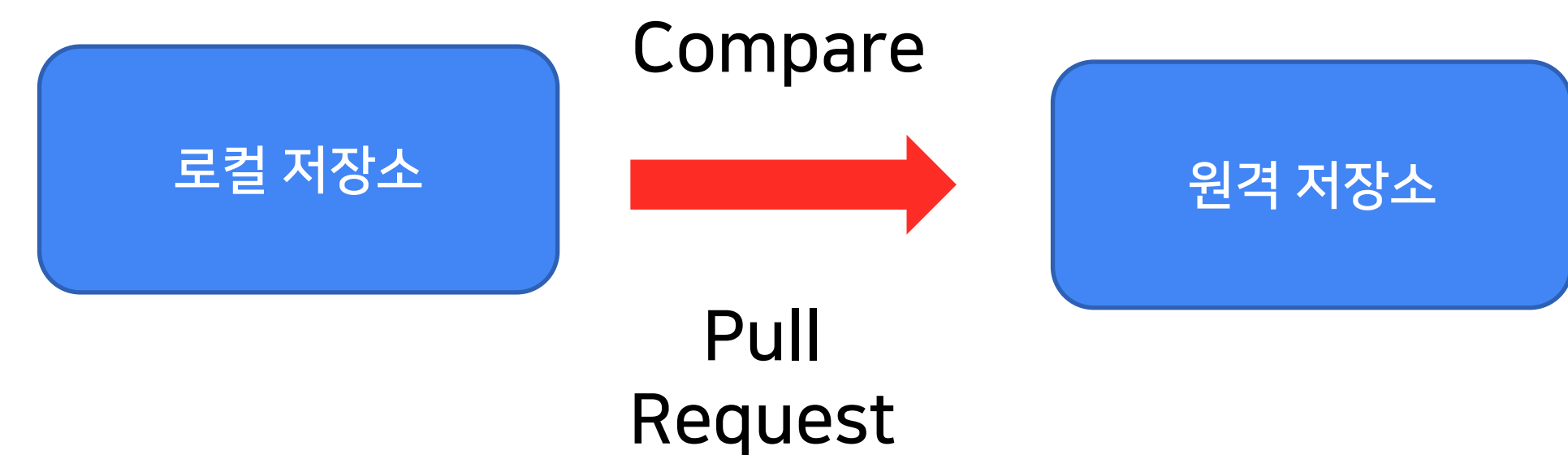
with Git Desktop



Git 홈페이지에 들어가서 새로 고침(F5)을 해줍니다.

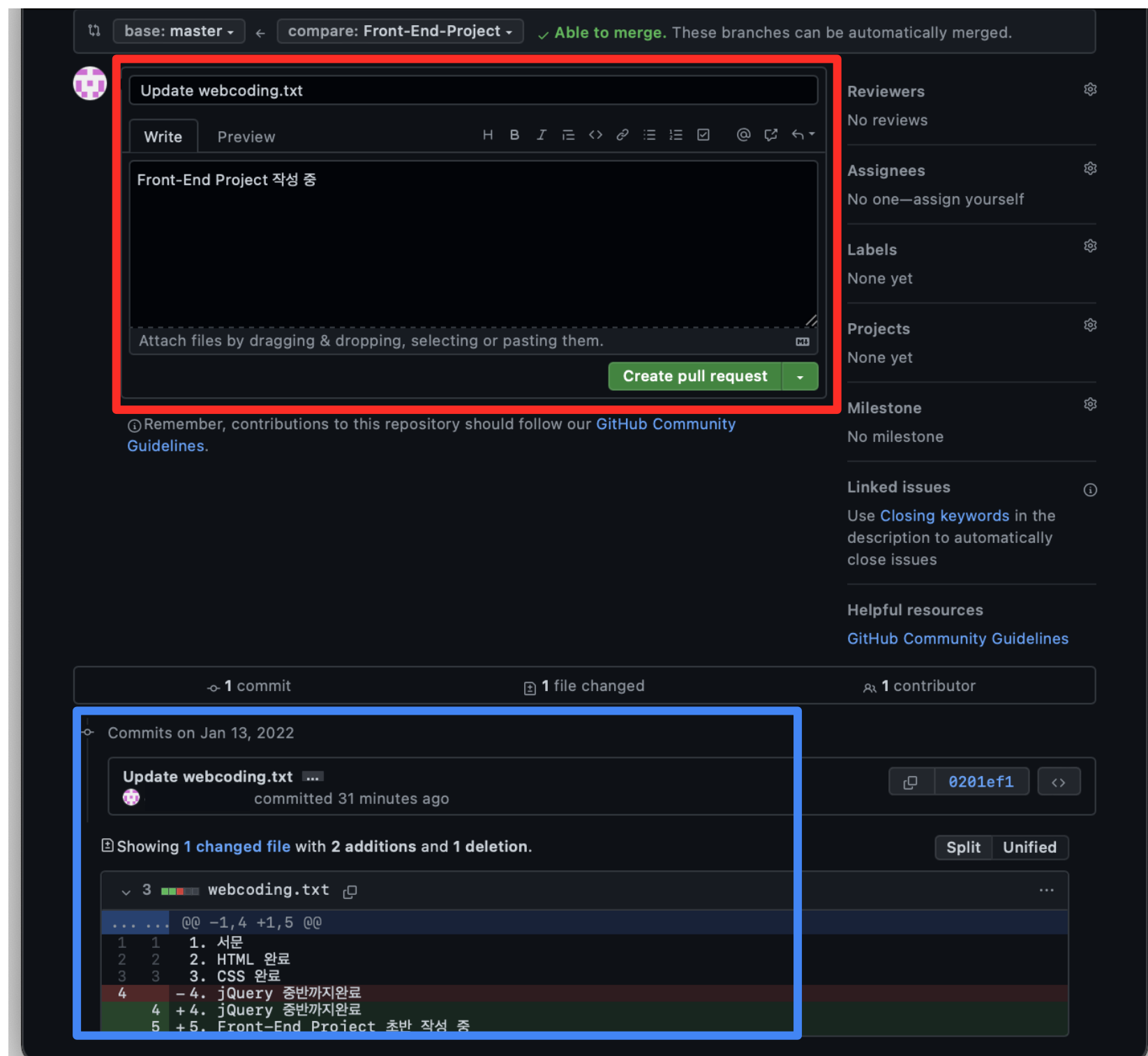
Compare (비교) & Pull Request (가져오기 요청)

즉, Front-End-Project branch를 master branch에 합치기 위한 과정입니다.



Git Branch 합치기

with Git Desktop



Git 홈페이지에 들어가서 새로 고침(F5)을 해줍니다.



정해진 규칙에 맞게 작성 후 Create pull request 버튼 클릭!

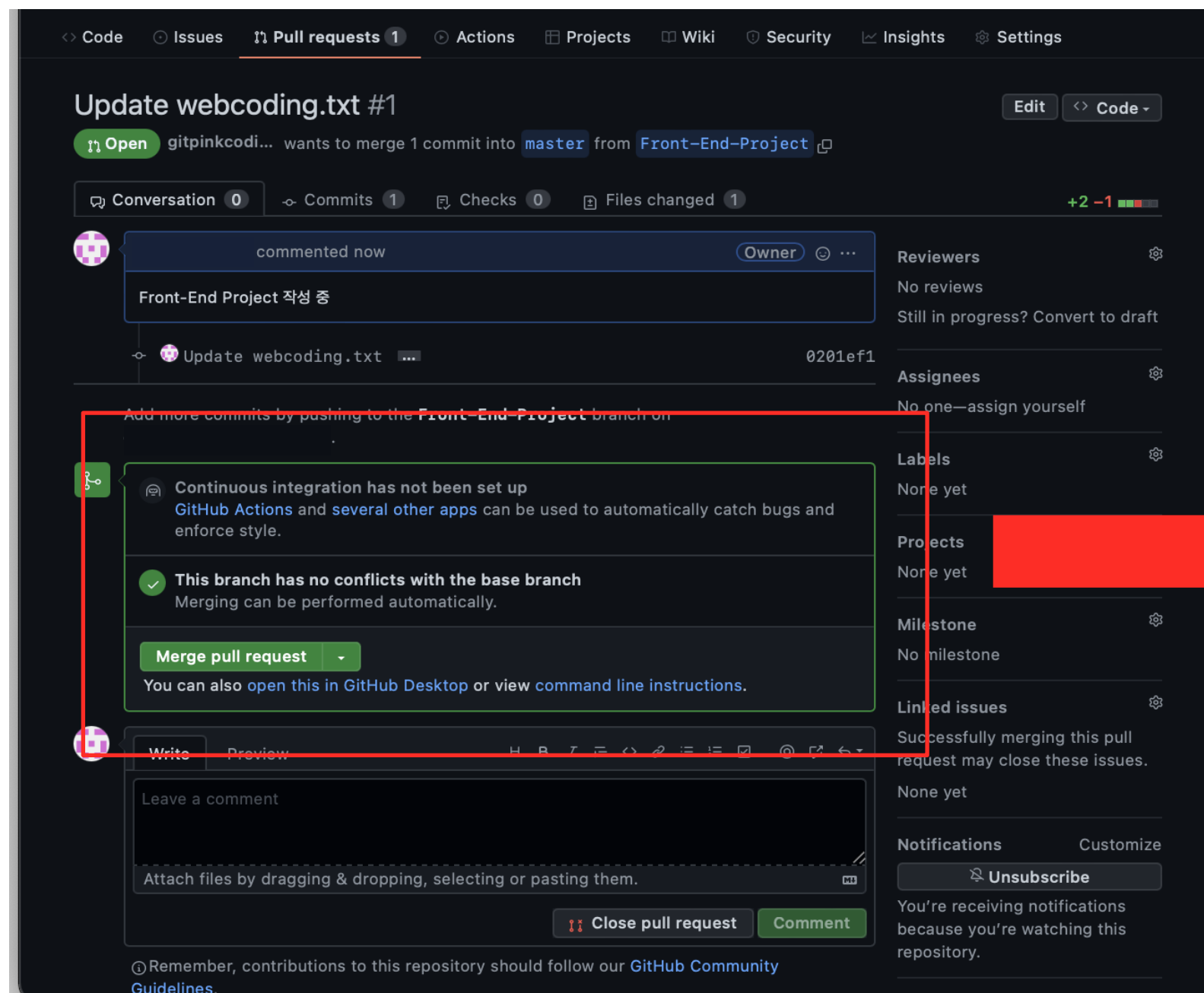


코드의 변경사항을 볼 수 있음



Git Branch 합치기

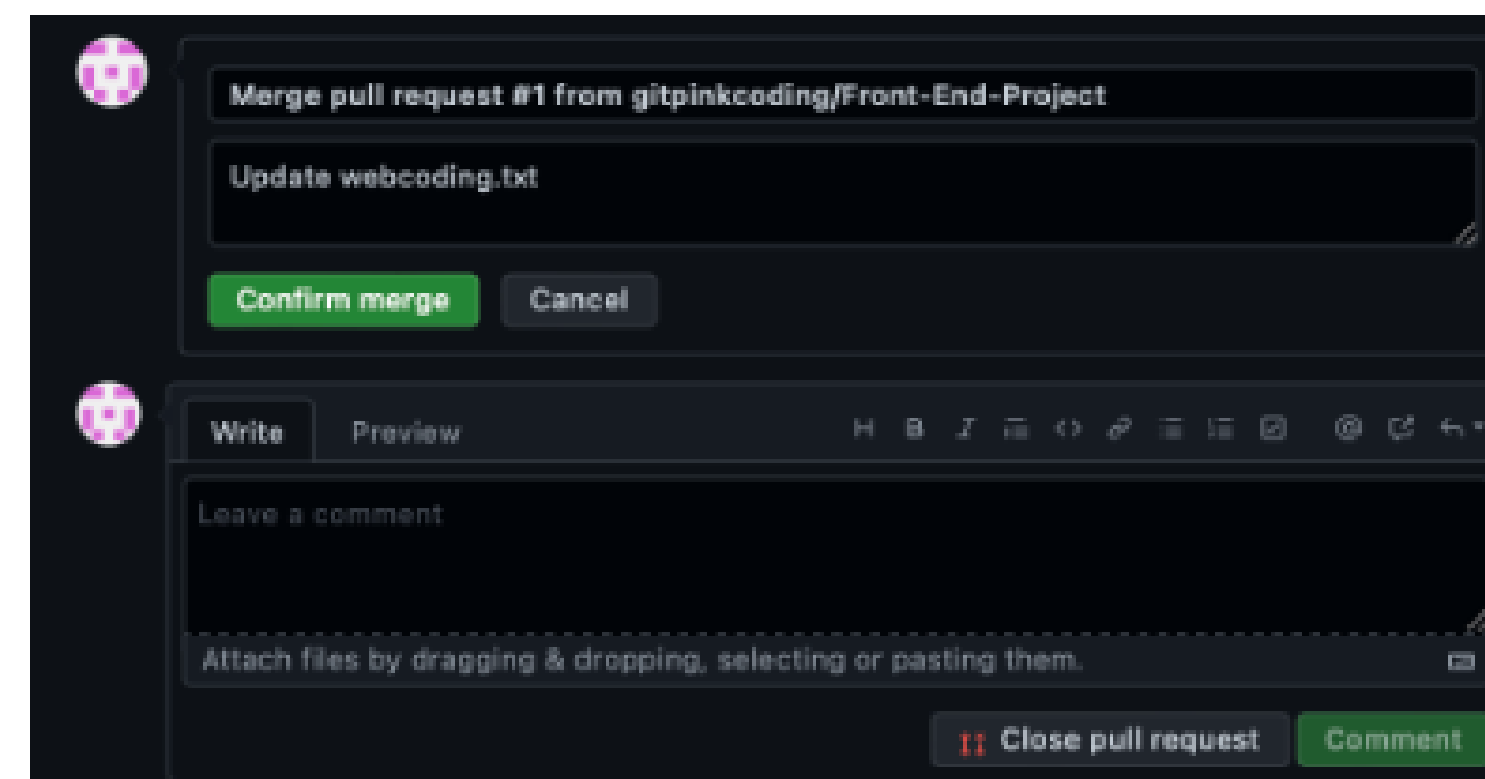
with Git Desktop



PR을 올린 직후 모습

This branch has no conflicts with the base branch의
의미는 충돌(conflicts)이 없다는 의미입니다.

다시 말해, Front-End-Project branch가 master branch에 합쳐지면서
코드상에 충돌이 없음을 의미합니다.



Confirm Merge를 눌러 Branch를 합치면 됩니다.

Git Branch 합치기

with Git Desktop

주의사항 !!!

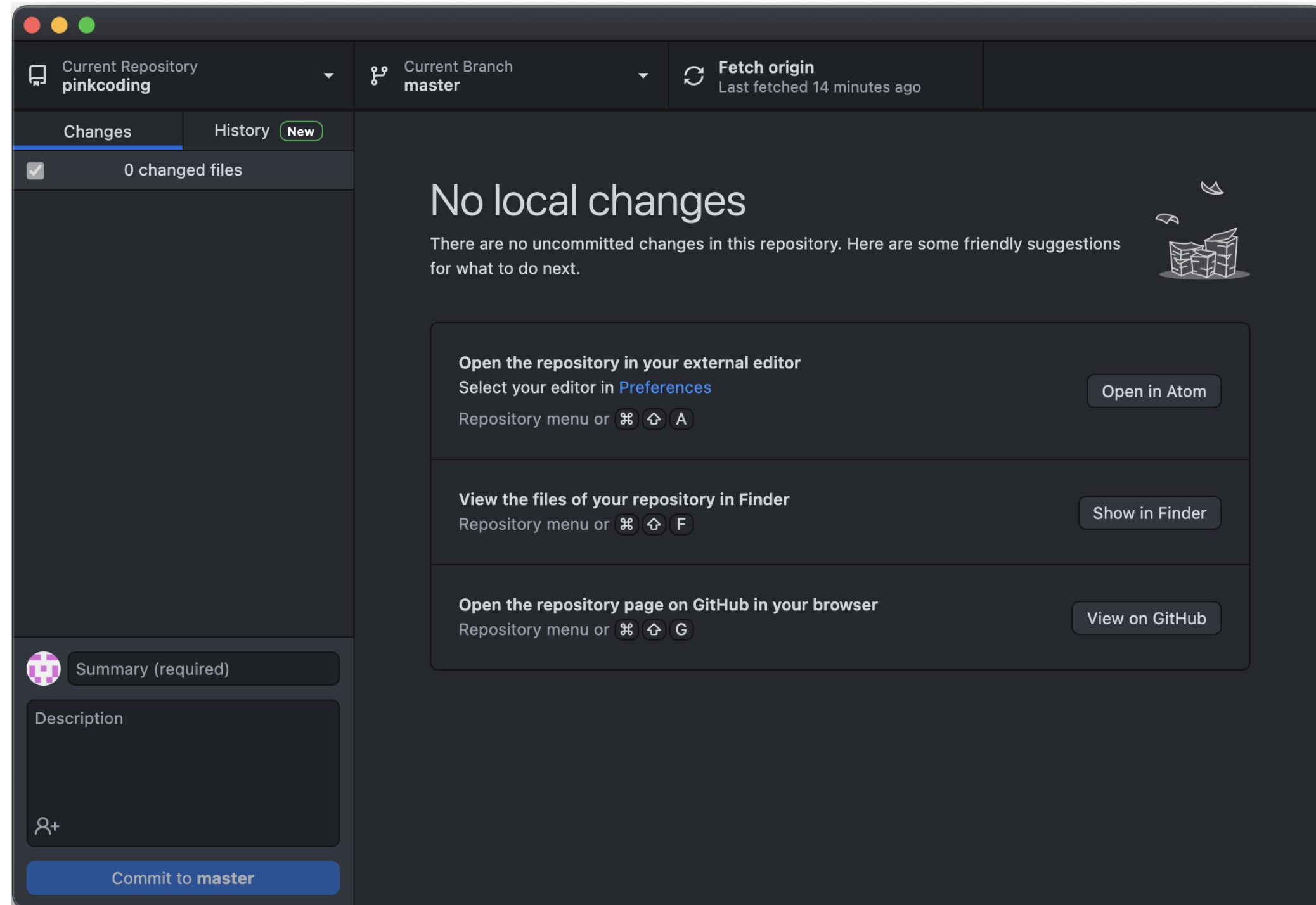
대부분의 상황에서 Merge는 신중 하게 작업이 되어야합니다.

따라서 별도의 권한을 가진 User가 Merge pull request를 클릭하여 관리 할것 입니다.

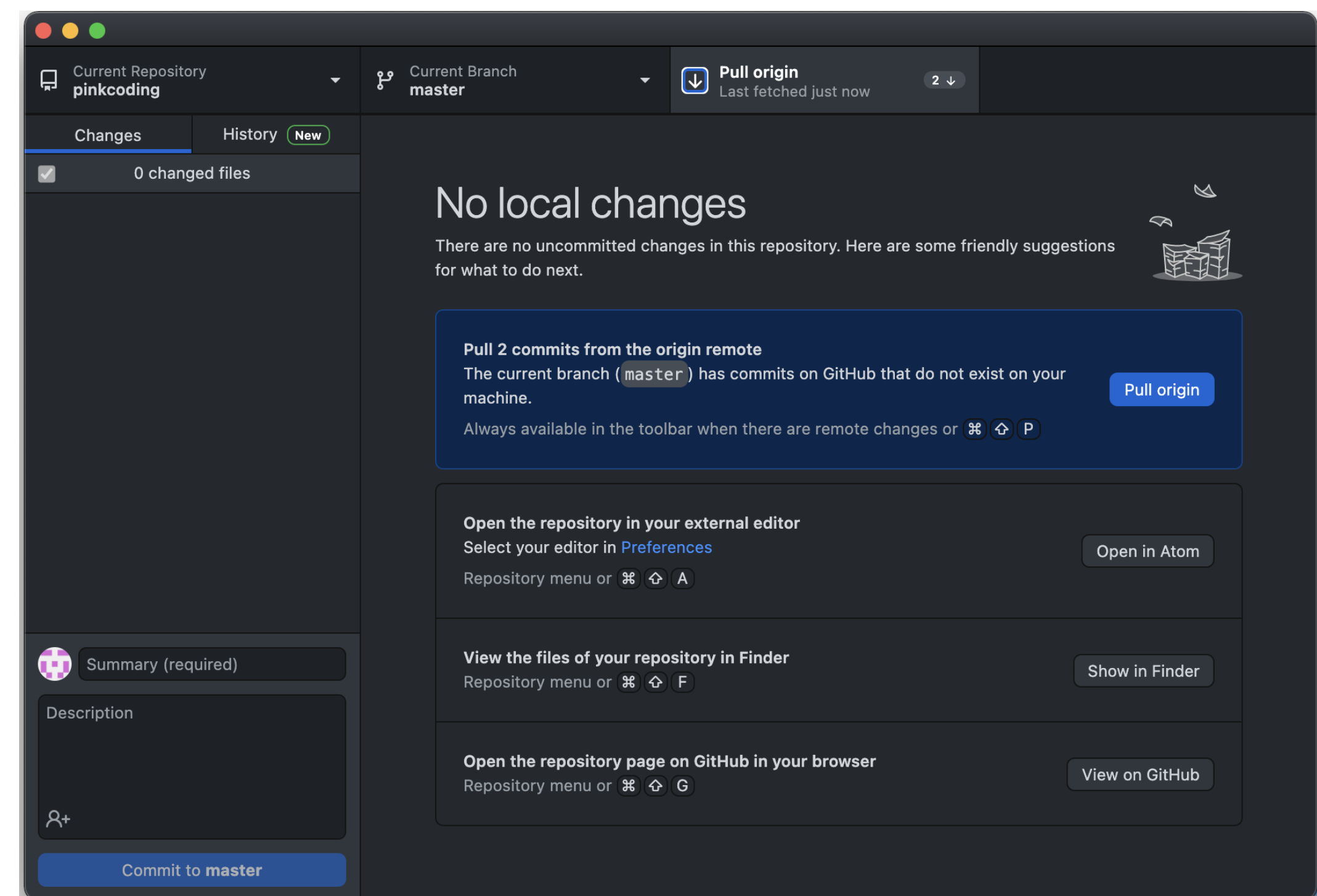


Git Branch 합치기

with Git Desktop



Merge가 잘 되었나 볼까요?
master branch로 checkout 후 fetch origin 클릭



Pull origin이 나왔군요!
git desktop이 갱신하라고 알려줍니다 어서 pull origi을 클릭합시다!

마무리

with Git Desktop

지금까지 Branch를 만들고 작업 후 원격 저장소(master)에
merge 하는 방법까지 (PR) 살펴보았습니다

좀 이해가 되었나요?

문의 / 제안 : [Slack](#) {@박정현, @이준원, @조영훈}

