

ILE3-030 GIT

코드의 관리 편의성을 위하여 형상관리도구인 GIT을 사용하는 방법에 대하여 학습

GIT

분산 버전 관리 시스템, 소스 코드 관리 시 사용

2005년 리누스 토발즈가 CVS 쓰다가 소스 코드를 날려먹고 1달만에 만듦

Local 저장소로 사용하거나, GitHub과 같은 Git Repository 제공자를 사용

버전 관리

- 누가 어떻게 변경했는지에 대한 추적
- 다수의 인원이 1개의 프로젝트에 투입 될 때 수정부분의 동기화
 - 문제 발생 시 복구
 - 프로젝트 진행 시 과거 시점에서의 RollBack이 필요할 때
 - 마이너 버전(Branch)를 이용한 기존 프로젝트 영향 최소화
 - 병합(Merge)를 통한 마이너 버전의 통합

기초 활용 방법

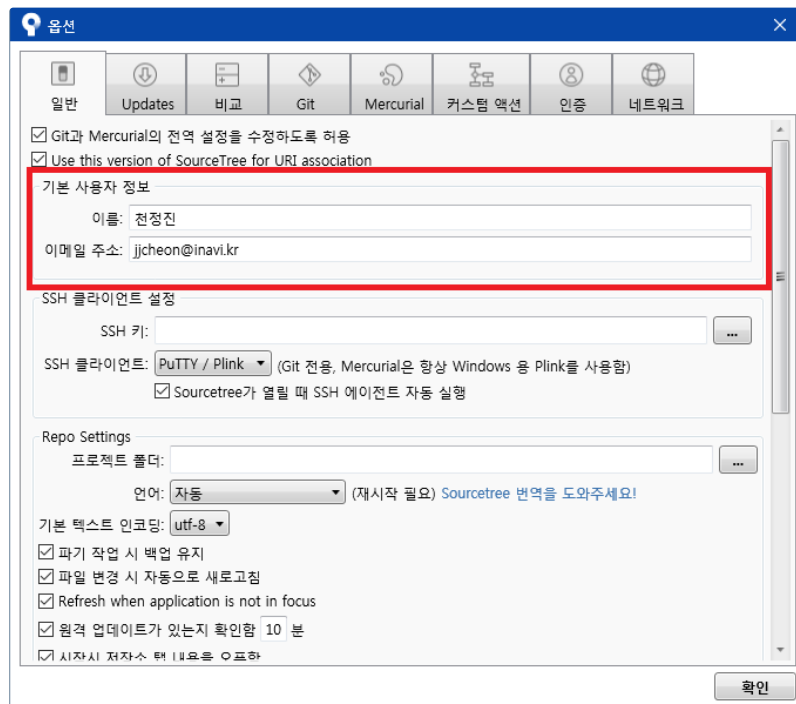
1) SourceTree 설치

1. SourceTree 설치

[SourceTree 홈페이지](#)에서 다운로드 후 설치

2. 개인 설정

커밋하는 사람의 정보를 개인 정보(이름 및 이메일) 설정도구 -> 옵션을 눌러 설정 페이지 진입 후 기본 사용자 정보 설정



The image shows the 'Options' dialog box in SourceTree. The 'General' tab is selected. The 'Basic User Information' section is highlighted with a red box. It contains fields for 'Name' (전정진) and 'Email Address' (jjcheon@inavi.kr). Below this, the 'SSH Client Settings' section shows 'SSH Key' and 'SSH Client' (PuTTY / Plink) with a checkbox for 'SourceTree가 열릴 때 SSH 에이전트 자동 실행'. The 'Repo Settings' section includes 'Project Folder', 'Language' (자동), and 'Encoding' (utf-8). At the bottom, there are checkboxes for 'Keep track of file changes', 'Refresh when application is not in focus', and 'Show commit history in the main window'.

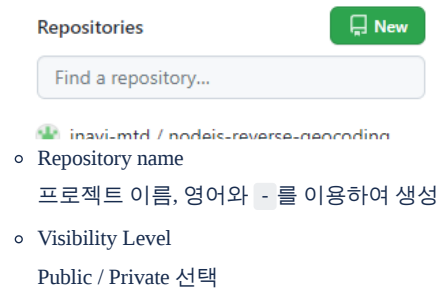
2) Repository 생성

1. Project list 페이지 진입

GitHub에 초기 로그인 시 좌측에 Repository 목록이 표시 됨

2. Project 생성

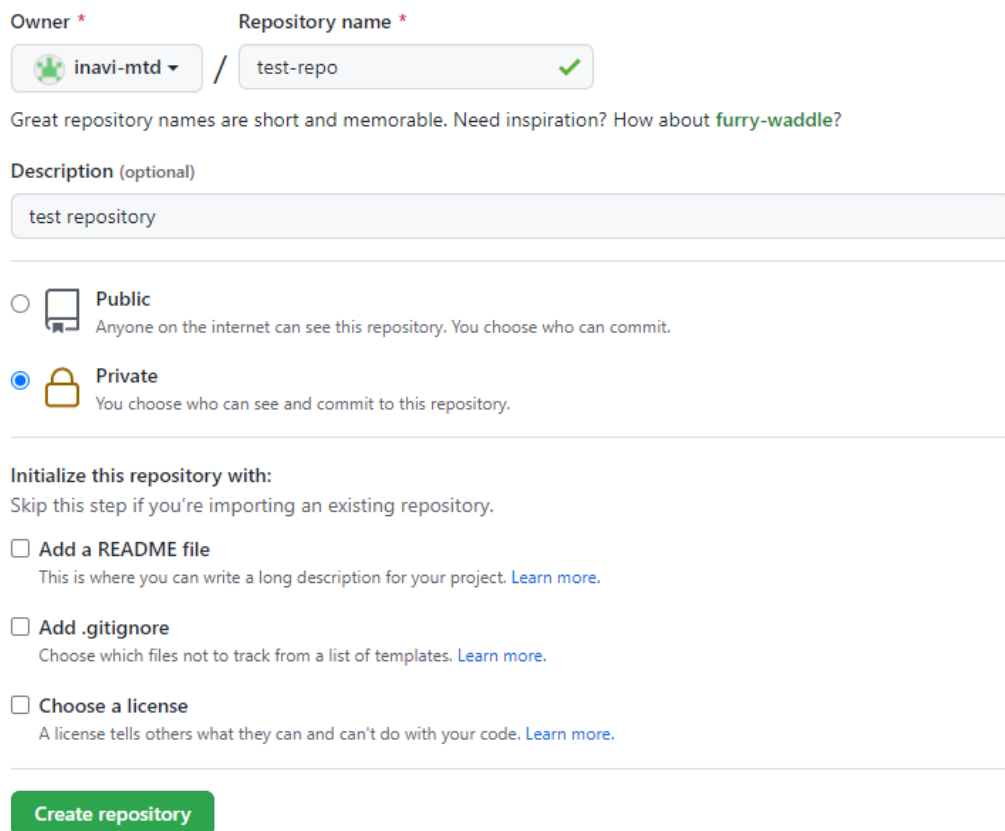
왼쪽 상단의 초록색 버튼 (New Project)를 클릭 후 생성 할 Repository의 정보 입력 후 하단의 초록색 버튼 (Create repository)를 클릭



The screenshot shows the GitHub interface for creating a new repository. At the top, there is a 'Repositories' header and a green 'New' button. Below this is a search bar labeled 'Find a repository...'. Under the search bar, a list of repositories is shown, with the first one being 'inavi-mtd / nodejs-reverse-geocoding'. Below the list, there are two bullet points: 'Repository name' and 'Visibility Level'. Under 'Repository name', it says '프로젝트 이름, 영어와 - 를 이용하여 생성'. Under 'Visibility Level', it says 'Public / Private 선택'.

Create a new repository

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



The screenshot shows the 'Create a new repository' form. It has two main sections: 'Owner' and 'Repository name'. The 'Owner' section has a dropdown menu with 'inavi-mtd' selected. The 'Repository name' section has a text input field with 'test-repo' and a green checkmark. Below these fields, there is a text input field for 'Description (optional)' with the value 'test repository'. Under the description field, there are two radio buttons: 'Public' and 'Private'. The 'Public' option is selected. Below the radio buttons, there is a section titled 'Initialize this repository with:' with three checkboxes: 'Add a README file', 'Add .gitignore', and 'Choose a license'. All three checkboxes are unchecked. At the bottom of the form, there is a green 'Create repository' button.

4) Checkout

Checkout은 서버에 저장되어 있는 Project를 Local로 복사 해 오는 작업

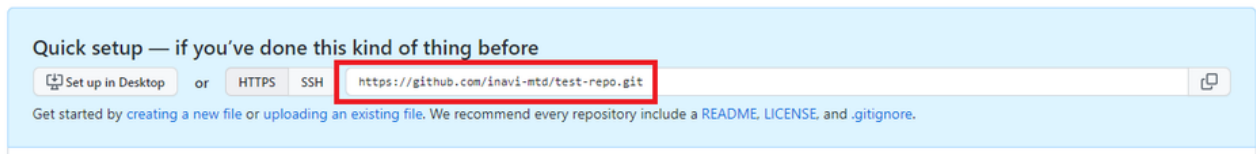
1. Project list 페이지에서 해당 Project 선택 후 진입



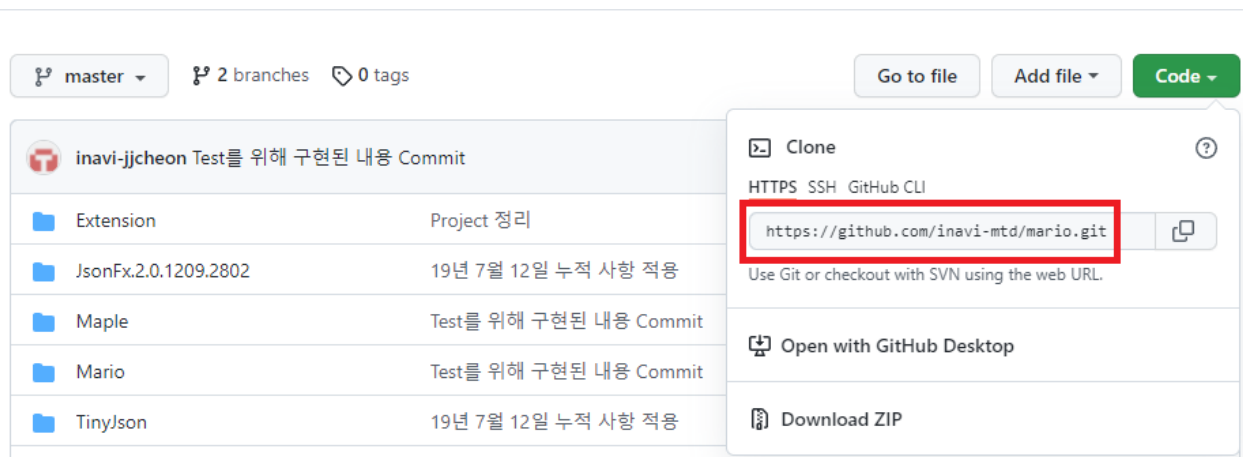
2. Project 페이지에서 URL 복사

Project Page에서 Repository URL을 확인

- Commit을 하나도 안 한 경우에는 Project Page에 URL이 표기 됨

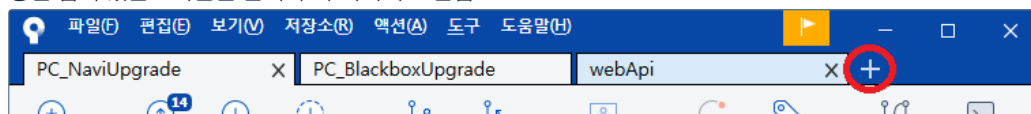


- Commit을 한 경우에는 좌측 상단의 Code 버튼을 클릭하면 URL이 표기 됨

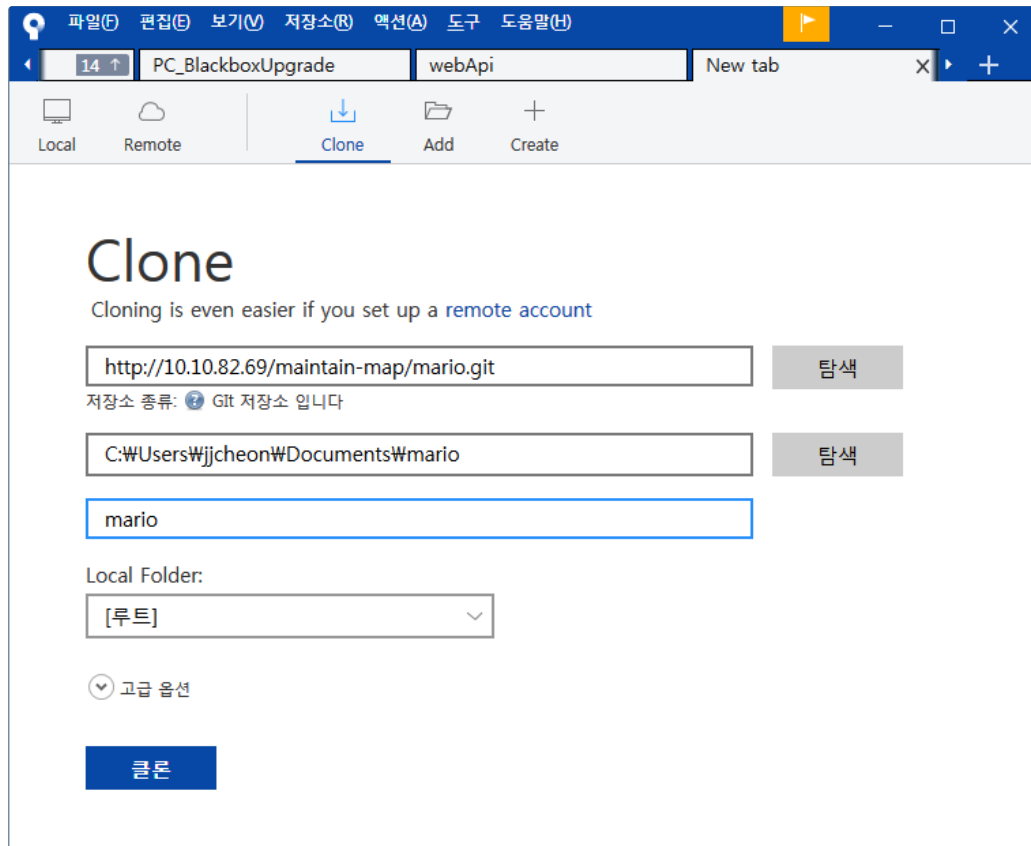


3. SourceTree에서 Project CheckOut

- a. 상단 탭에 있는 + 버튼을 눌러 추가 페이지로 진입

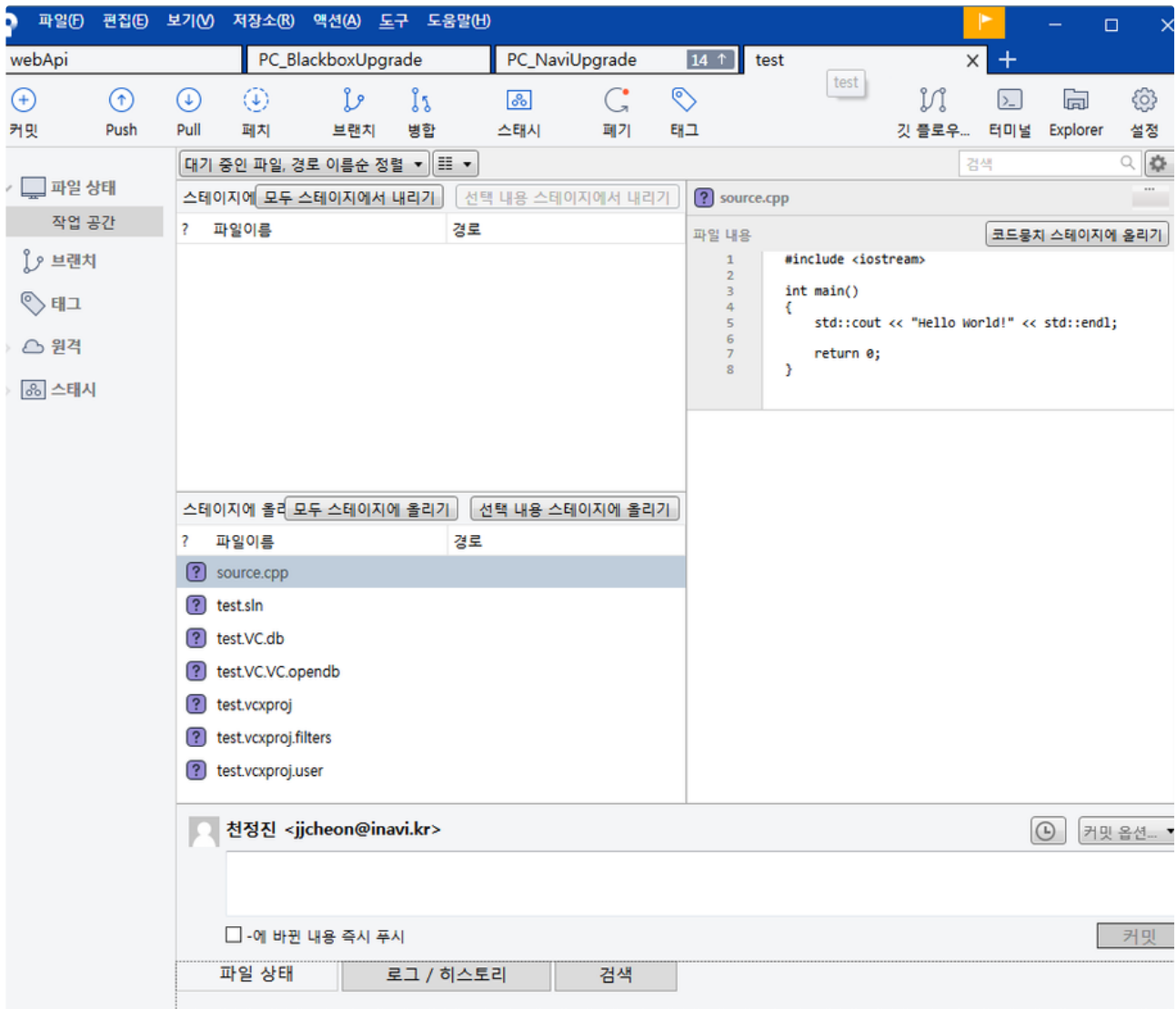


- b. 맨 위의 텍스트 박스에 URL을 복사하면 자동으로 아래 칸들이 채워짐
사용자 입맛에 맞게 수정 후 아래의 클론 버튼 클릭



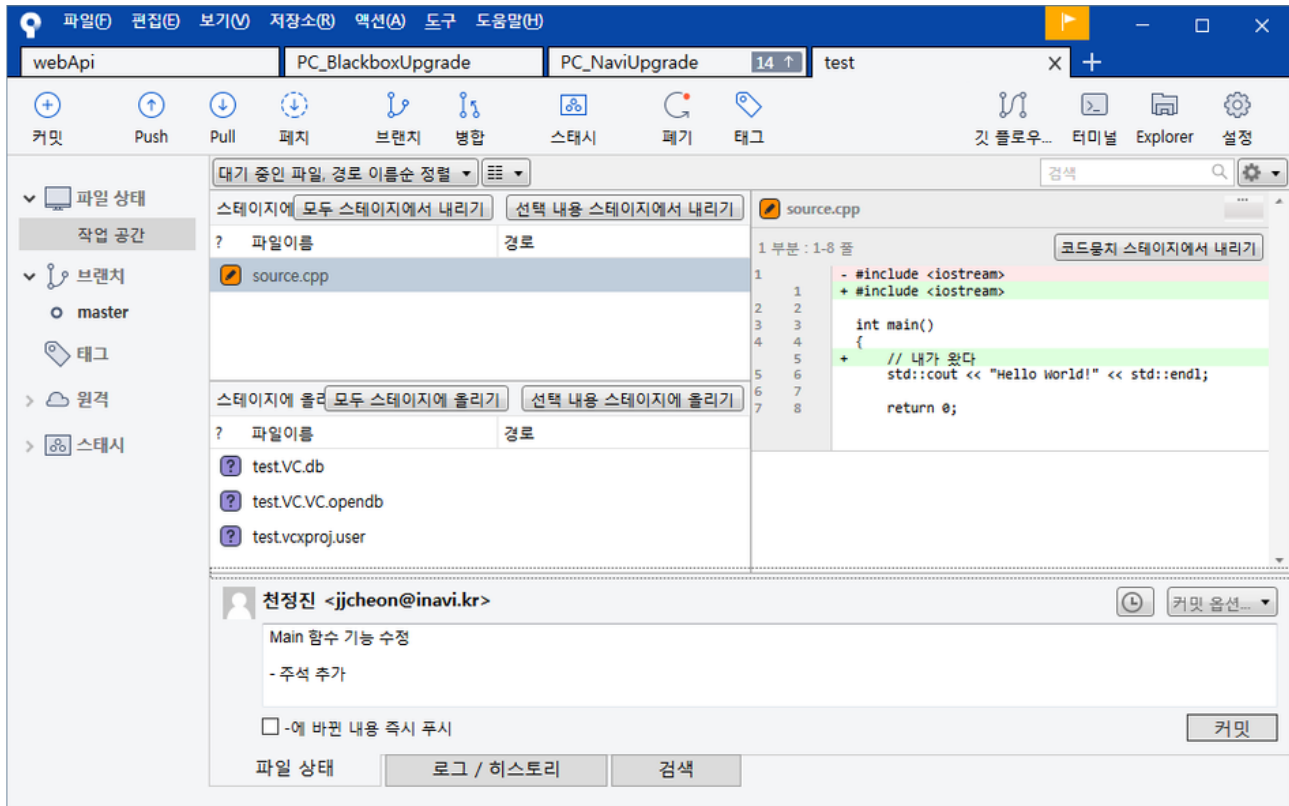
5) Commit

수정 내역이 있을 시 아래와 같이 SourceTree의 파일 상태 탭에서 변화 정보가 표시 됨

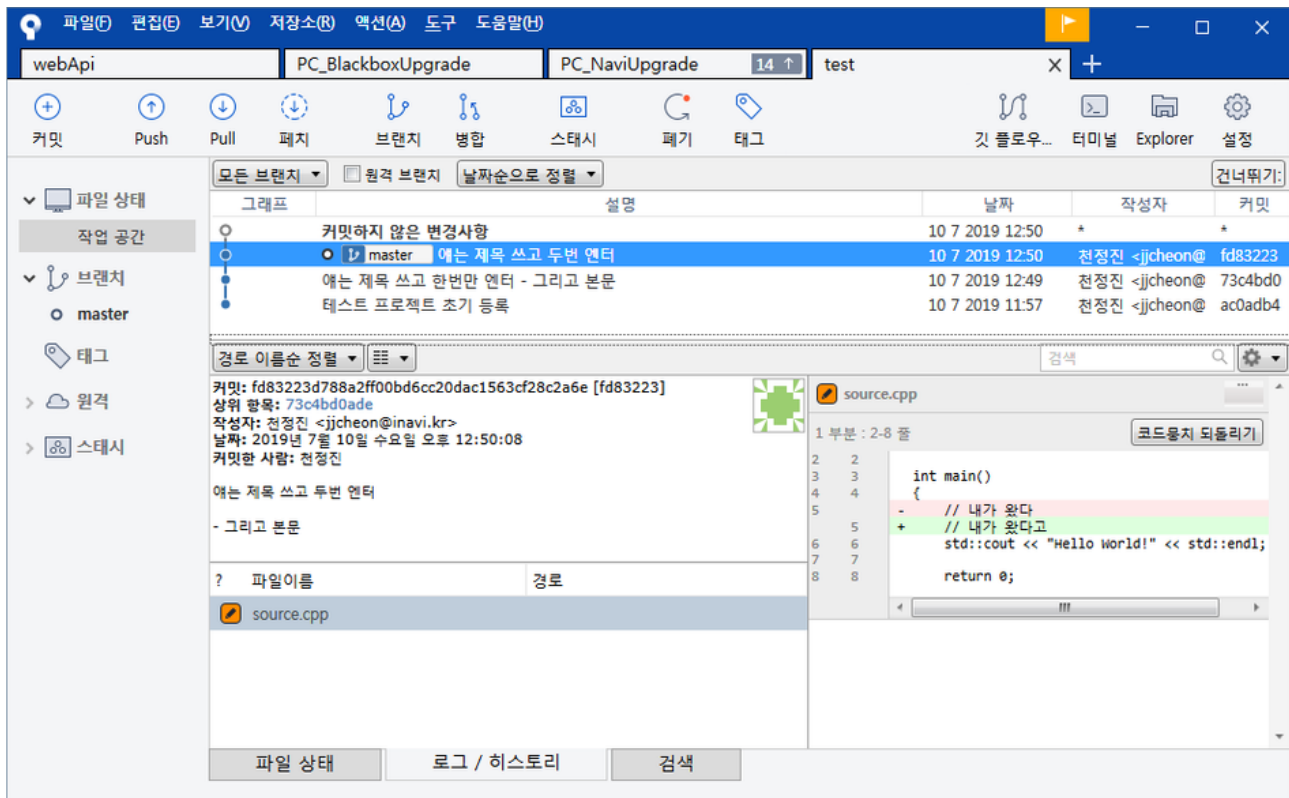


중간의 파일 목록에서 저장할 내역을 스테이지에 올리기 버튼을 통해 위로 올리고 하단에 변경 내역에 대한 로그를 기록 함

로그 기록 시 첫번째 줄에는 제목을 기록하고 두 번 줄바꿈을 한 후 상세 내용을 적을 것



변경 사항 기록 시 줄바꿈을 한 번 한 것과 두 번 한 것은 이후 히스토리를 볼 경우 다음과 같이 차이가 발생한다
한번만 줄바꿈을 한 것은 본문이 히스토리 내용에 나오는 것을 볼 수 있다

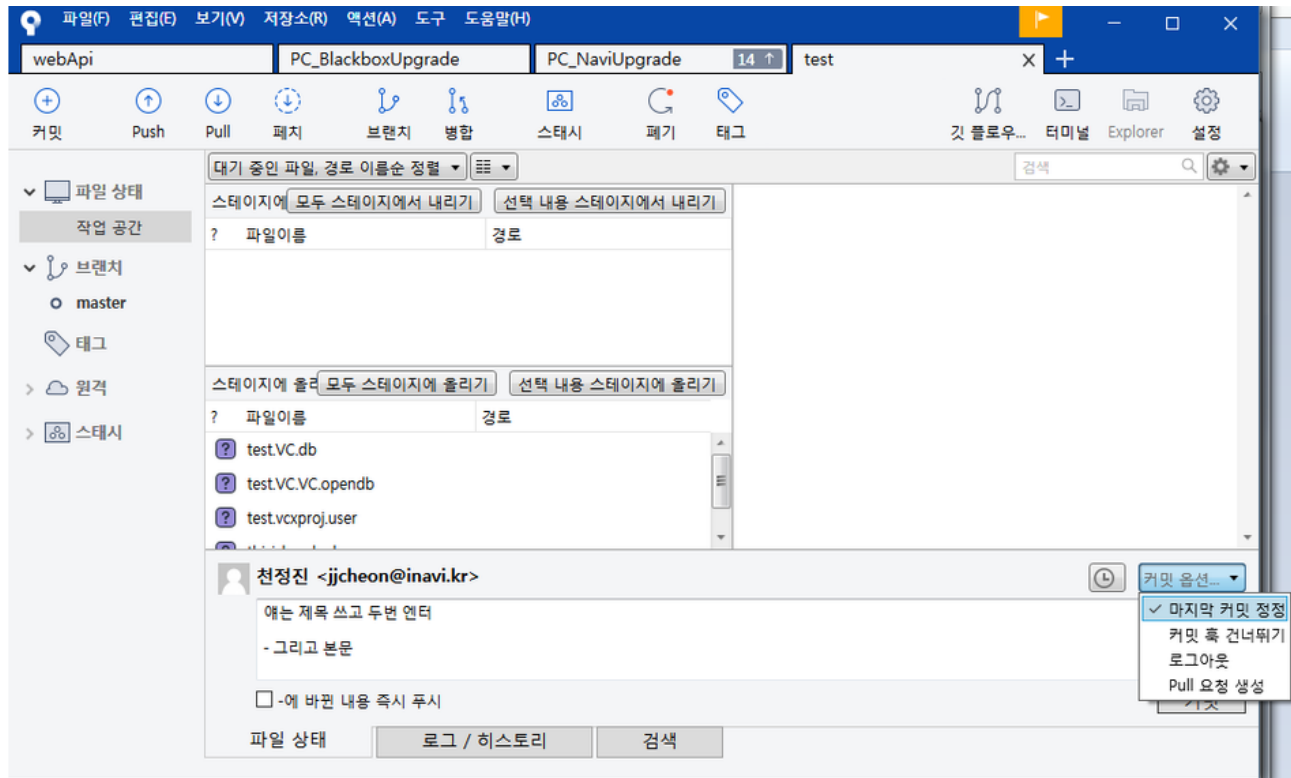


6) Commit 수정

Commit을 했는데 변경 사항을 잘 못 기입했거나 넣어야 하는 파일을 빼먹은 경우가 있을 수 있다. 이때 마지막 커밋에 대한 수정 기능을 통해 변경 사항 수정 및 파일 추가가 가능하다

마지막 커밋 수정을 하고 싶을 때는 오른쪽 하단의 커밋 옵션 버튼을 누르고 마지막 커밋 정정을 누른다.

마지막 커밋 정정을 누르면 변경 사항 정보가 자동으로 마지막 커밋 내용으로 변경되고, 변경 사항 정정 및 변경 파일 추가를 수행 후 커밋 버튼을 누르면 된다



7) Pull

Pull을 수행하면 서버에서 최신 정보를 가져오게 된다.

Commit을 하기 전 반드시 Pull을 수행하여 서버에서 최신 정보를 가져올 것

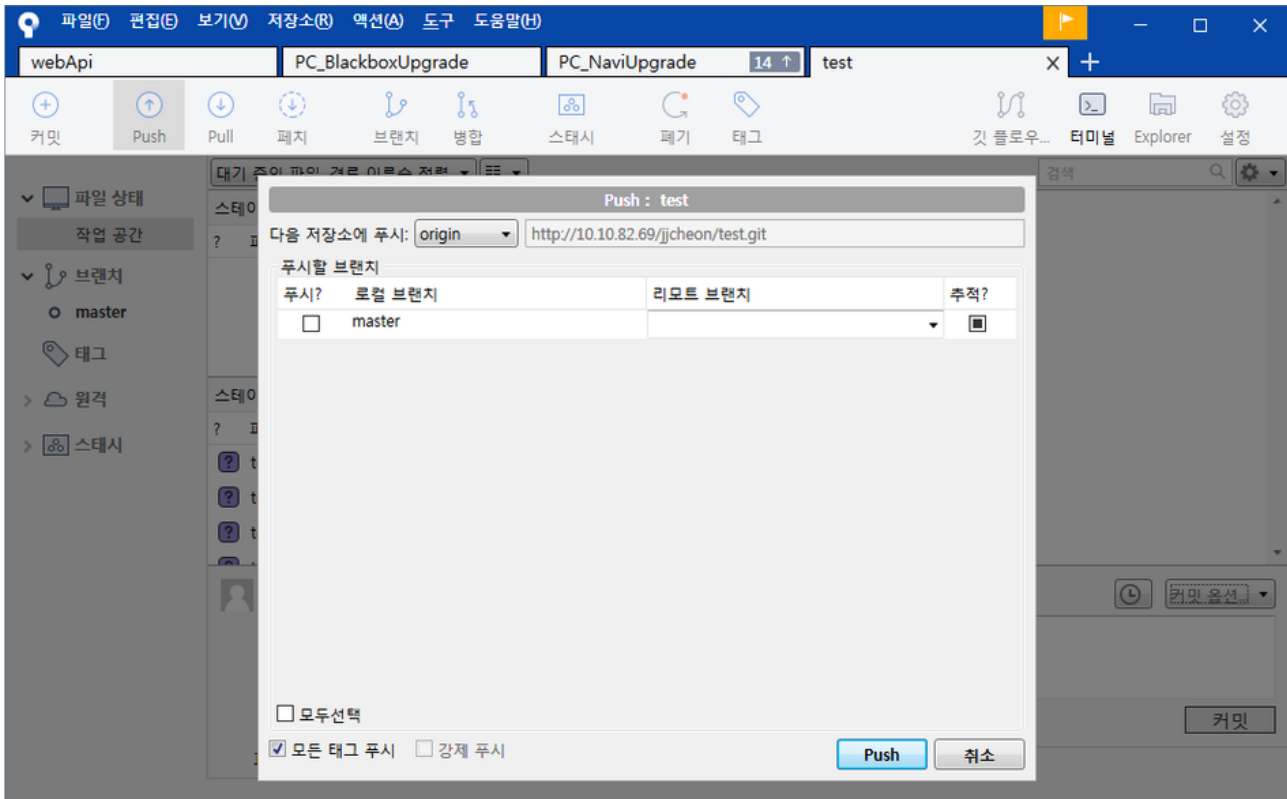
8) Push

Push를 수행 시 Commit한 데이터를 서버에 업로드 하게 된다.

Pull과 동일하게 Push를 하기 전 Pull을 수행해서 서버에서 최신 정보를 가져와 코드가 꼬이지 않도록 조심 할 것

Push 버튼을 누르면 아래 그림과 같이 어떤 브랜치를 서버에 업로드 할 지 선택을 하는 창이 뜨게 된다.

현재 자신이 작업 중인 브랜치를 서버의 어떤 브랜치에 적용할 지 선택 후 Push를 누르면 된다.



9) Stash [🔗](#)

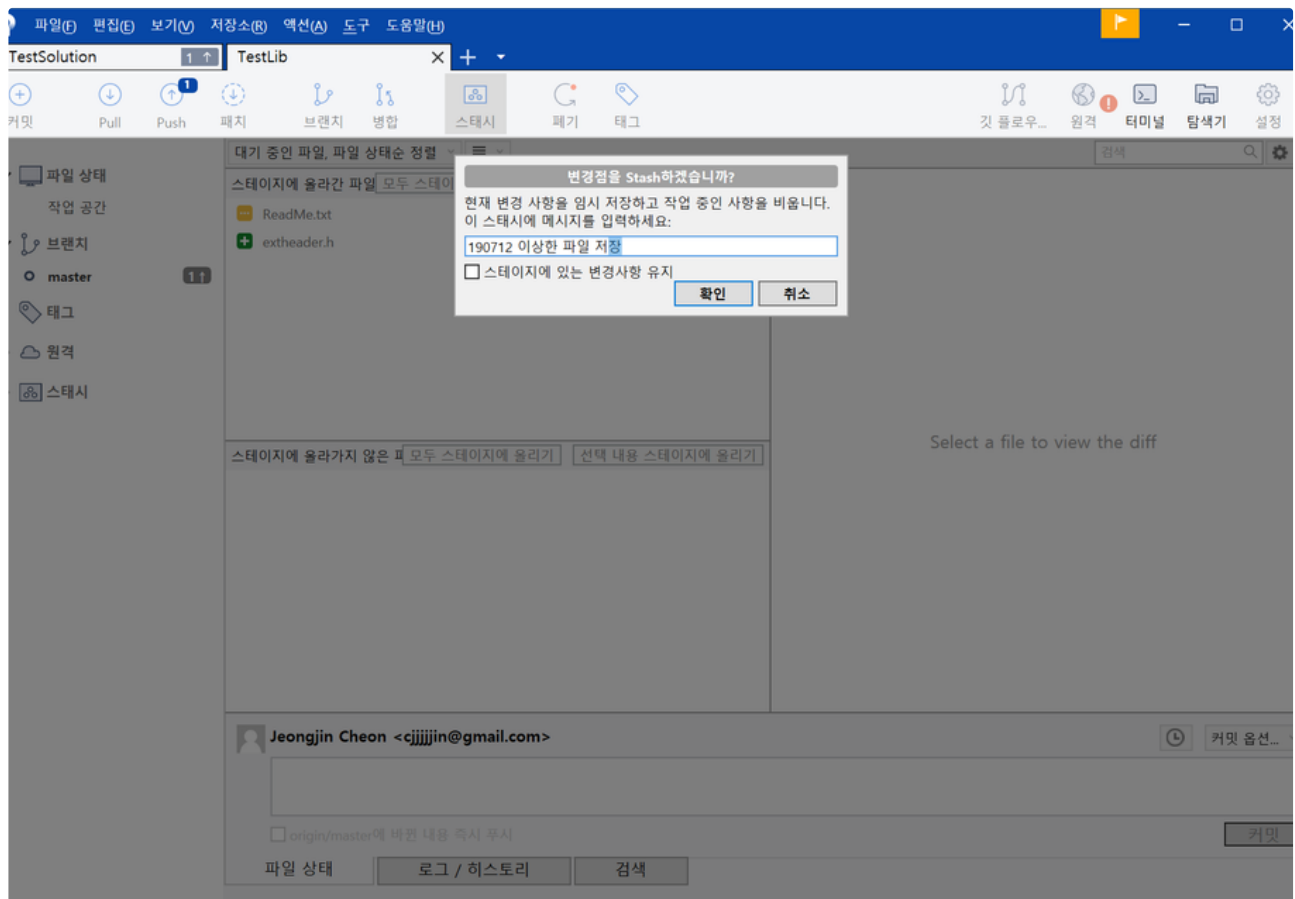
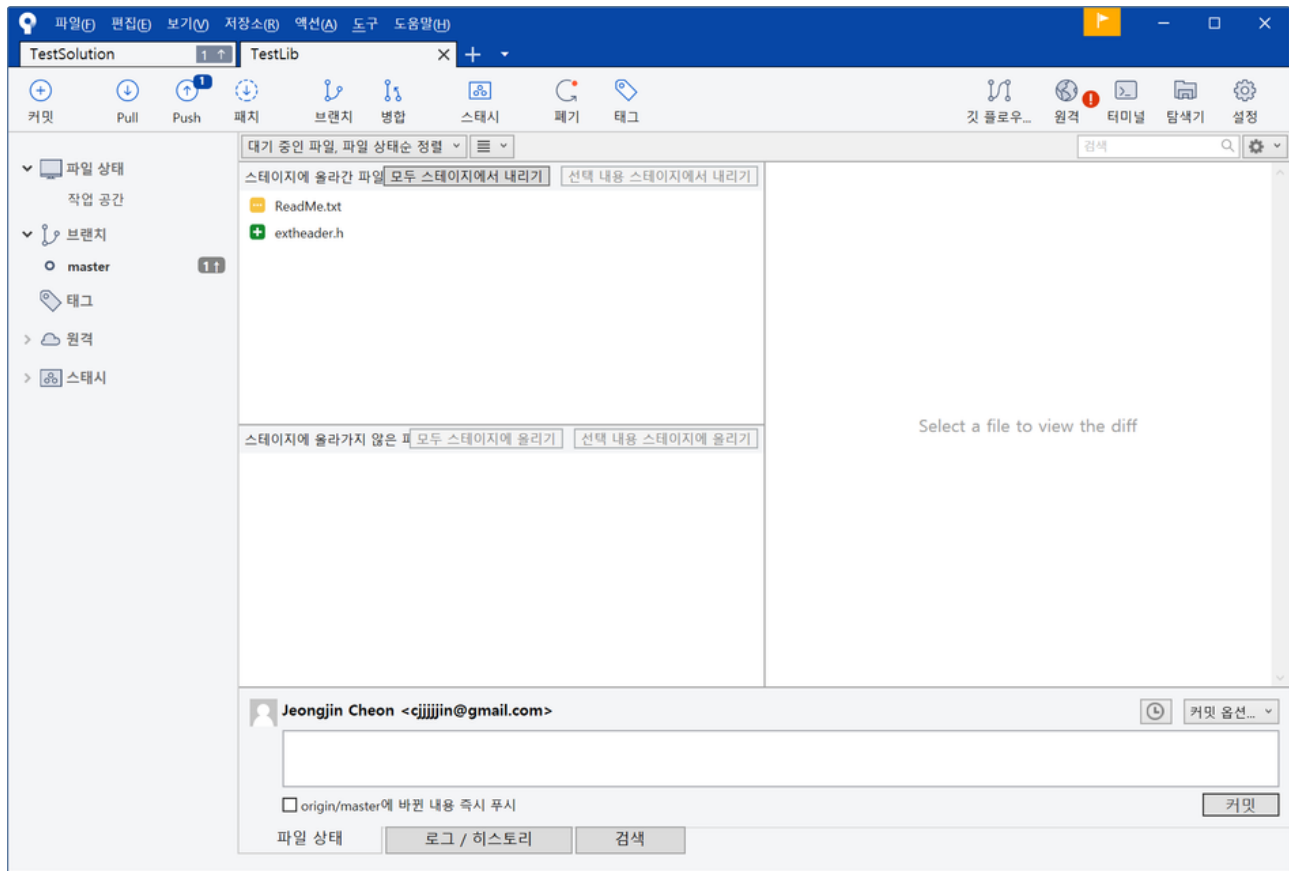
수정내역이 있을 때 브랜치 이동 또는 과거 Commit으로 이동을 시도 하였을 때 수정내역으로 인하여 이동이 불가능 한 경우가 있다.

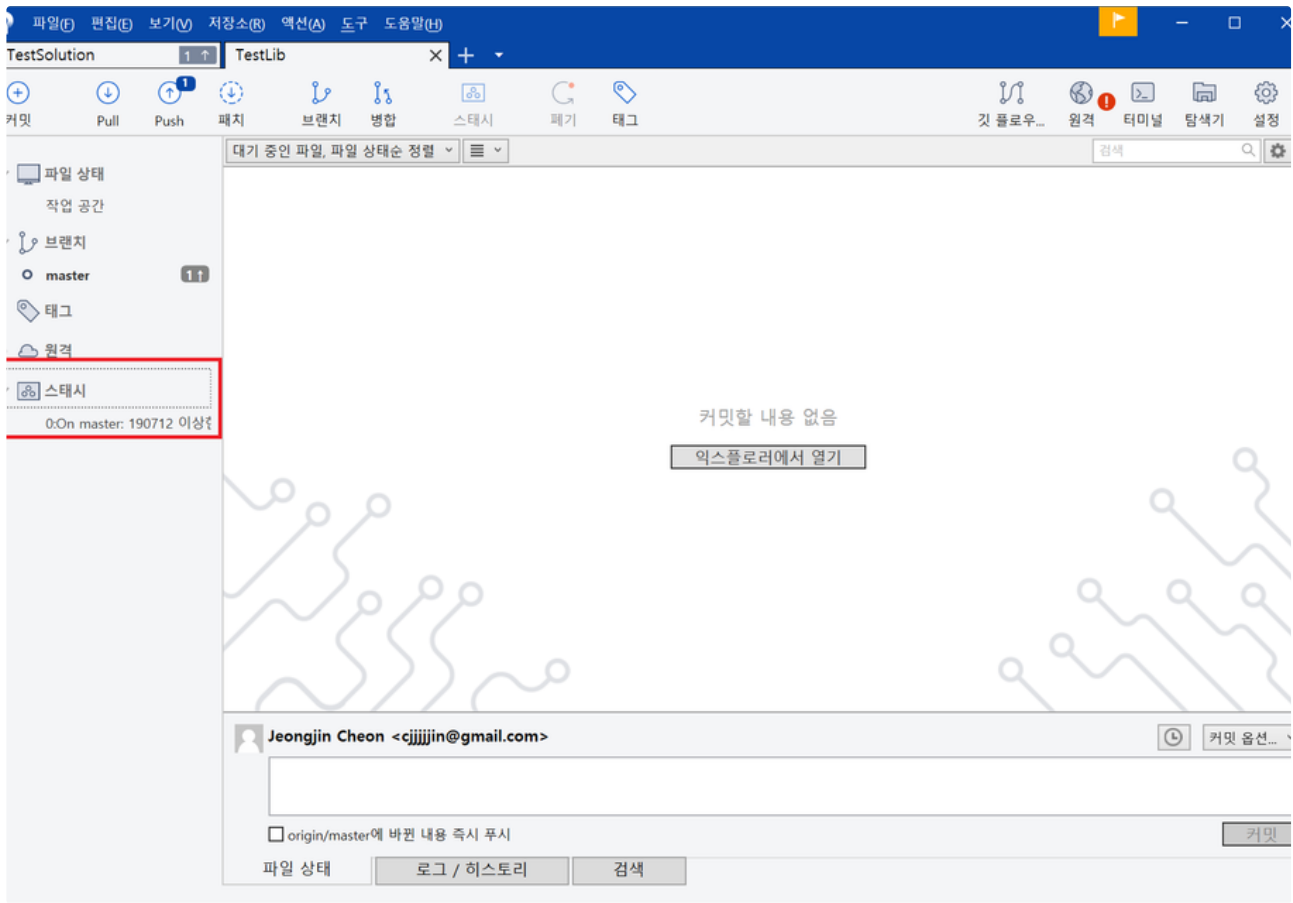
Stash는 수정 내역을 임시적으로 저장한 후 Clean을 수행하는 작업을 말한다. Stash 한 내역은 언제든지 다시 복구가 가능하다

1) Stash 수행 [🔗](#)

변경 사항의 경우 자동으로 Stash 처리가 되나, 추가 내역의 경우 자동으로 Stash가 안된다. 이 때 Stash에 추가하고 싶은 내역이 있을 경우 해당 파일을 스테이지로 올린 후 Stash를 수행하면 된다

Stash를 생성하는 경우 해당 Stash에 대한 명칭을 기록하여 주어야 한다





2) Stash 적용 [🔗](#)

좌측의 Stash 항목에서 적용 할 항목을 오른쪽 클릭 후 적용을 클릭 할 경우 해당 수정 사항이 적용 되게 된다.

