**Github Desktop 사용자 가이드**

|  |  |
| --- | --- |
| **작성자** | **이우형** |
| **법인** | **아이라이브** |
| **부서** |  |
| **작성일** | **2020.05.25** |
| **버전** | **1.0** |

**개정이력**

| 버전 | 작성일 | 변경내용 | 작성자 | 승인자 |
| --- | --- | --- | --- | --- |
| 1.0 | 2020.05.25 | 초안 작성 | 이우형 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

내용

[1. 개요 4](#_Toc42506439)

[2. Git의 사용법 4](#_Toc42506440)

[3. GitHub 로그인 하기 4](#_Toc42506441)

[4. GitHub Desktop 옵션 설정 5](#_Toc42506442)

[5. 저장소(Repository) 가져오기(Clone) 6](#_Toc42506443)

[6. 브랜치(Branch) 9](#_Toc42506444)

[7. 커밋(Commi) 하기 11](#_Toc42506445)

[8. 병합(Merge) 하기 13](#_Toc42506446)

[9. 푸시(Push) 하기 14](#_Toc42506447)

1. ****개요****

Git은 코드의 변경사항을 추적하는 시스템으로 코드가 언제, 어떻게 누구에 의해 변경 되었는지

기록하는 것이며, Github는 이러한 변경사항들을 클라우드에 올리는 웹사이트 입니다.

소스를 사용하다 보면 아래와 같은 문제로 업무의 연속성이 끊기는 경우가 발생하는데

이러한 문제를 Github Desktop을 통해 아주 간단하고 편리하게 사용 및 관리할 수 있습니다.

1) 소중한 자신의 소스 코드가 손상되거나 손실 되는 경우

2) 외부에서 자신의 소스 코드를 열람해야 하는 경우

3) 소스 코드의 변경된 부분을 찾아야 할 경우.

1. ****Git의 사용법****

A. Repository : 소스를 저장하는 폴더

- Repository를 컴퓨터에 저장하면 Git은 파일을 체크하고 변경사항을 추적 합니다.

B. Commit : 파일 변경 기록

- 파일의 생성이나 변경사항의 내용을 Git에 기록하는 것이 Commit 입니다.

C. Branch : 처음 기본으로 생성되는 것은 master이며, master branch에는 사용자의 모든

Commit이 반영되어 있습니다.

- master branch는 마지막 작업 결과물이라고 할 수 있습니다.

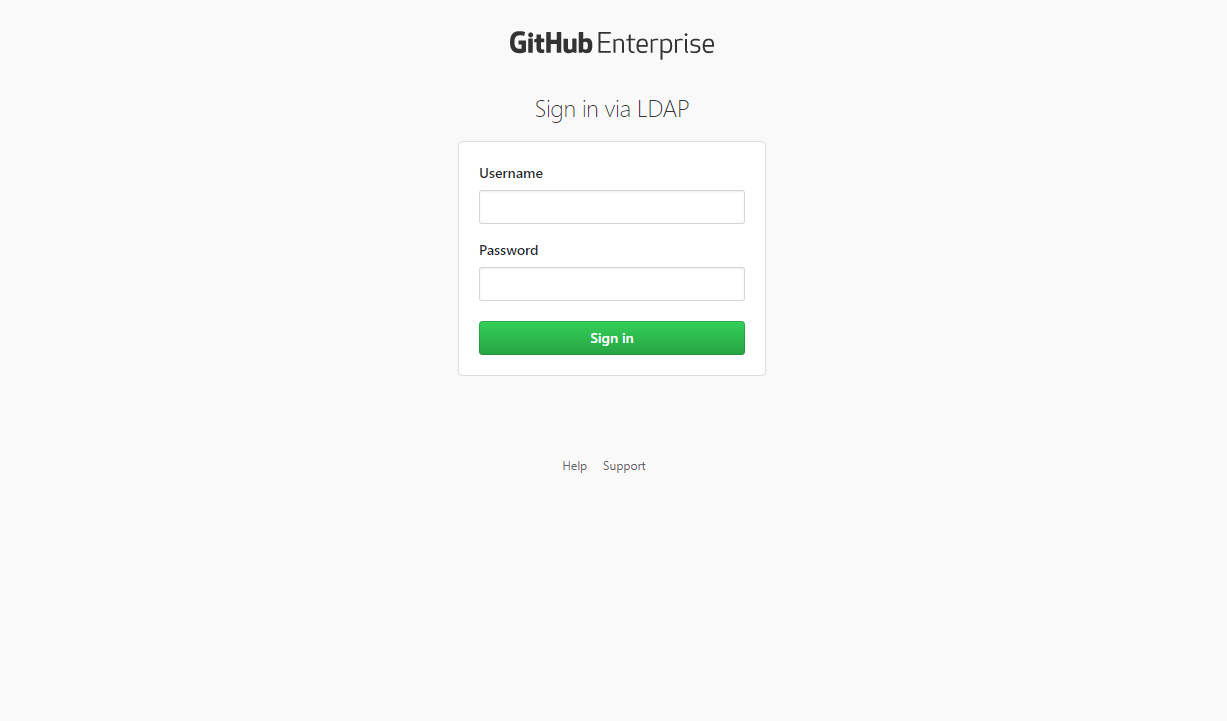
예를 들면 업무로 인하여 새로운 기능을 추가하거나 변경사항을 적용해야 할 때

운영 소스인 master branch를 건드리지 않고, 새로운 Branch를 생성하면 master와 똑같은

소스코드를 가진 Branch를 만들어 개발 및 테스트를 진행 할 수 있습니다.

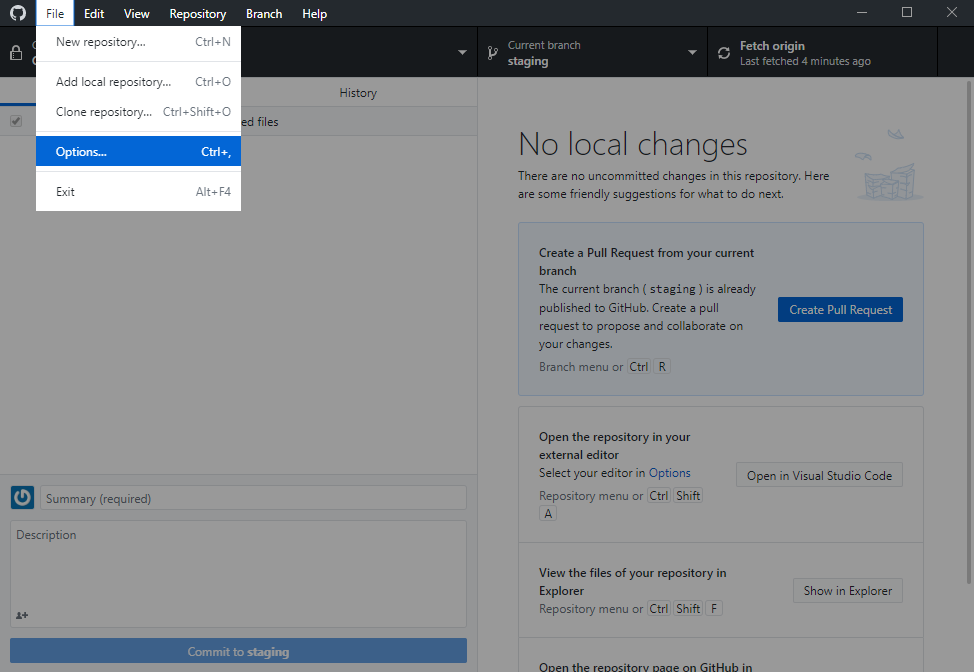
1. ****GitHub 로그인 하기****

A. 스마일게이트 Github 웹사이트에 로그인 합니다. (https://github.smilegate.net/login)

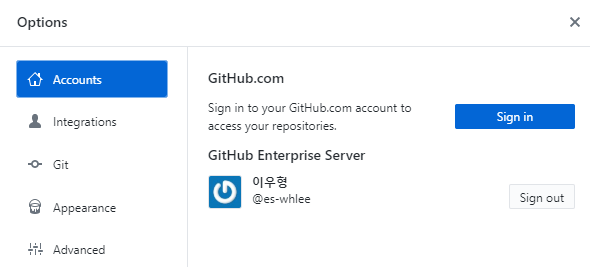


1. ****GitHub Desktop 옵션 설정****

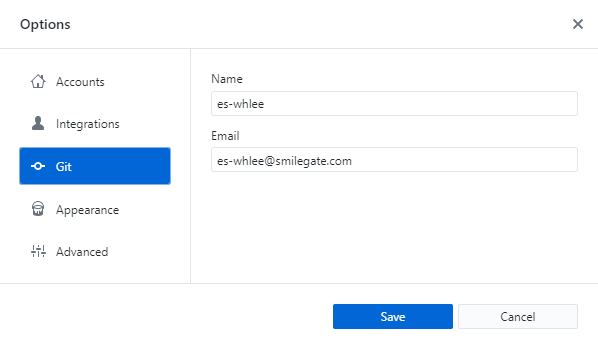
프로그램 상단 메뉴에서 “File > Options..”를 선택하여 옵션들을 확인 할 수 있습니다.



A. Accouts에서 Github와 연결된 계정(ad)을 확인 할 수 있습니다.



B. Git 에는 프로그램 설치 시 입력한 정보가 저장되며 여기 입력된 사용자 정보가

Git 작업 시 표기되므로 Accounts에 입력한 정보와 동일하게 만들어 줍니다. **(중요)**

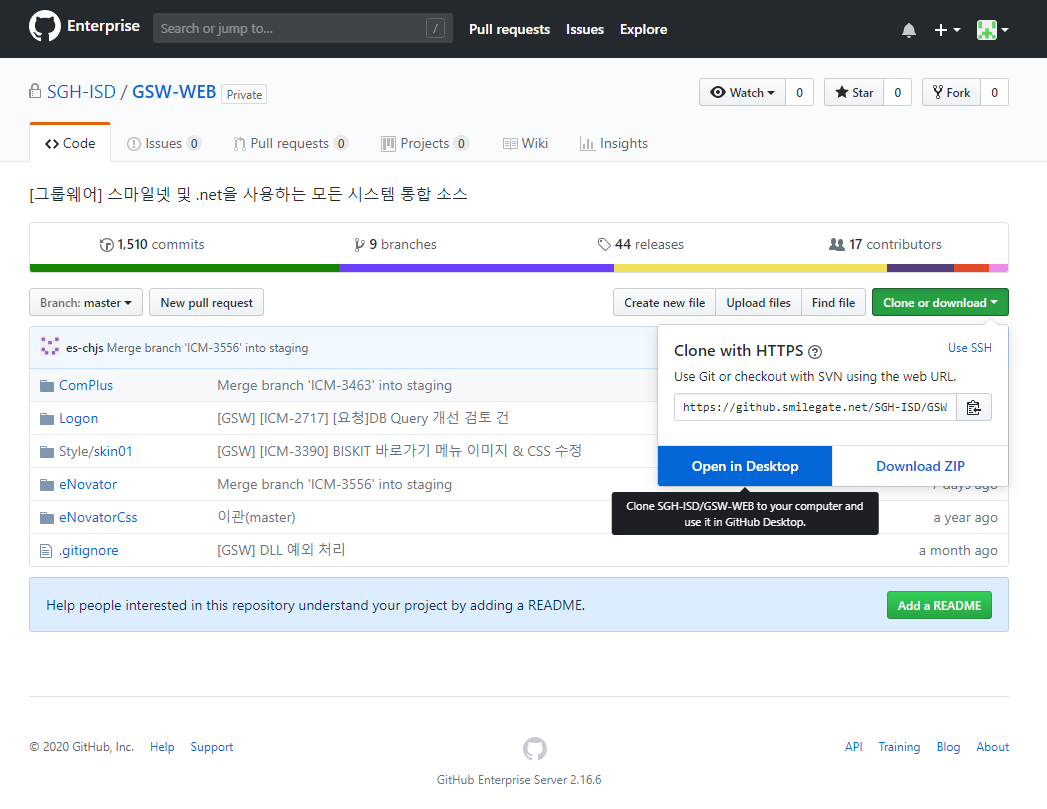
1. ****저장소(Repository) 가져오기(Clone)****

Github Desktop에서 Repository를 사용하기 위해서는 Github 서버에 업로드 되어있는 저장소(GSW-WEB) 소스파일을 “Clone” 기능으로 가져와야 하며 방법은 두가지 입니다.

A. Github 웹사이트에서 Clone 하기

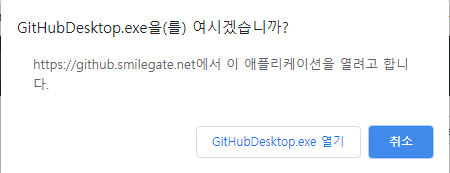
(1) "Clone or download > Open in Desktop" 을 순서대로 누릅니다.

**※ 이 작업으로 Repository URL 정보를 알 수 있습니다.**

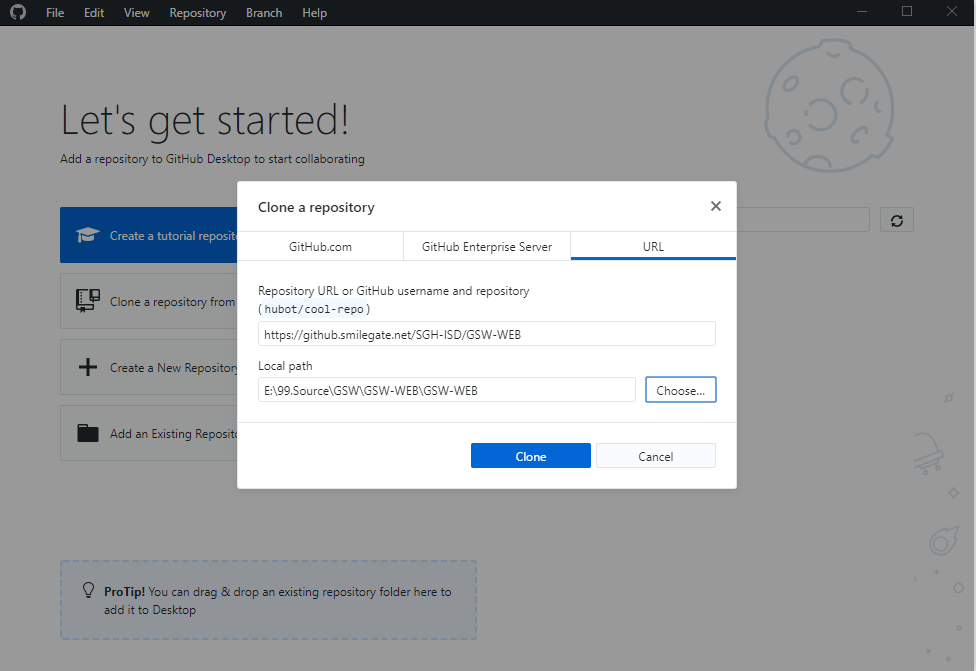


(2) GithubDesktop 프로그램을 실행 여부를 묻는 팝업이 활성화 되면 “GitHubDesktop.exe열기”를

클릭 합니다..

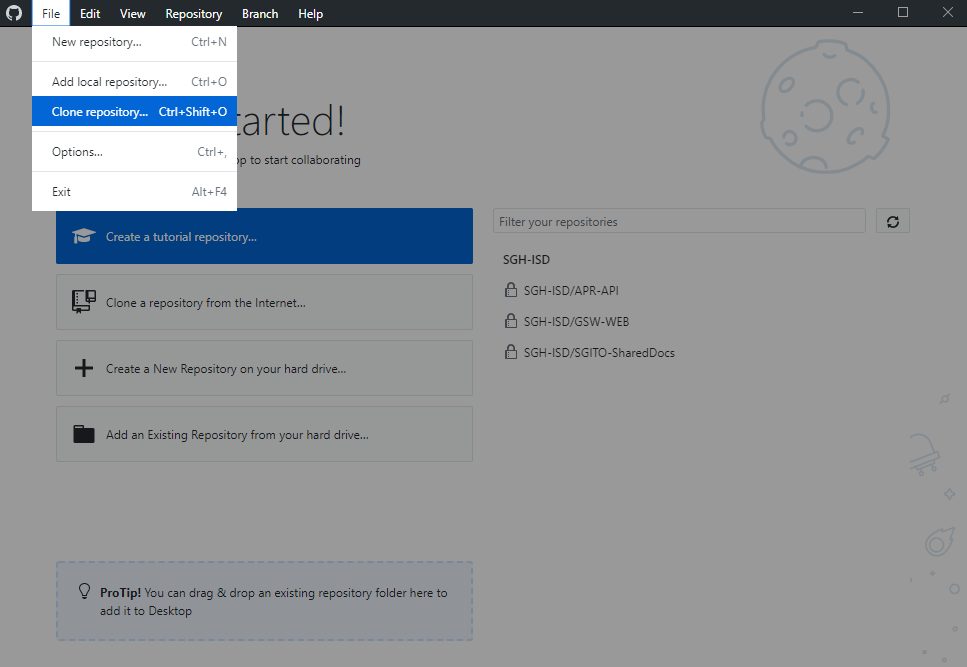


(3) “Repository URL or Github username and repository”에 항목에 Clone할 URL이 기입되었는지 확인하고, Local Path에는 본인 PC와 Github Repository를 연결 할 경로를 설정한 후 Clone을 클릭합니다.



B. Github Desktop에서 Clone 하기

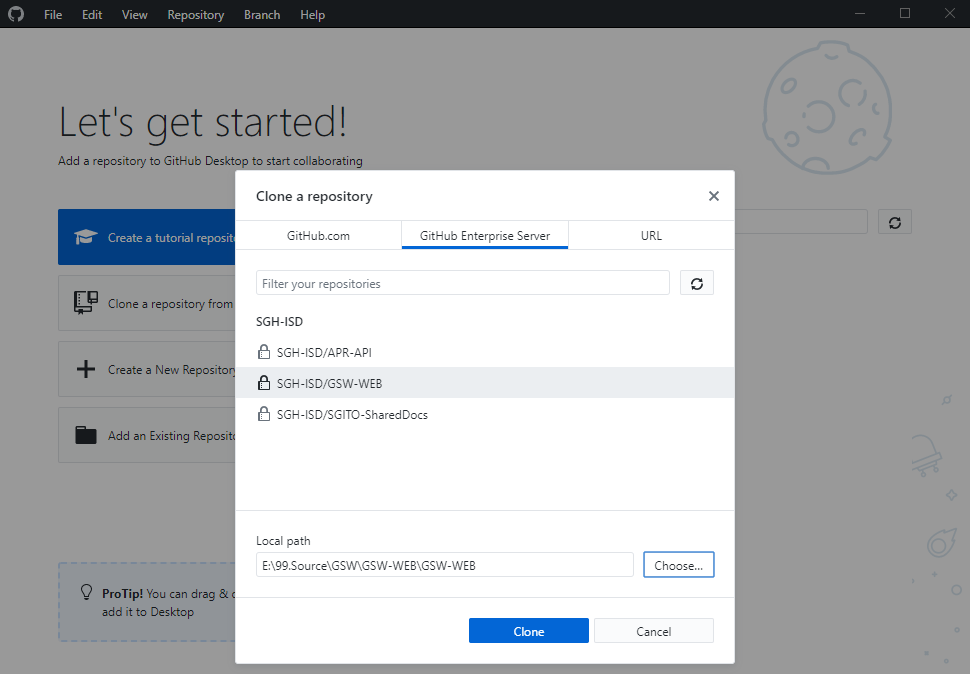
(1) 프로그램 상단 메뉴에서 “File>Clone repository..”를 순서대로 클릭 합니다.



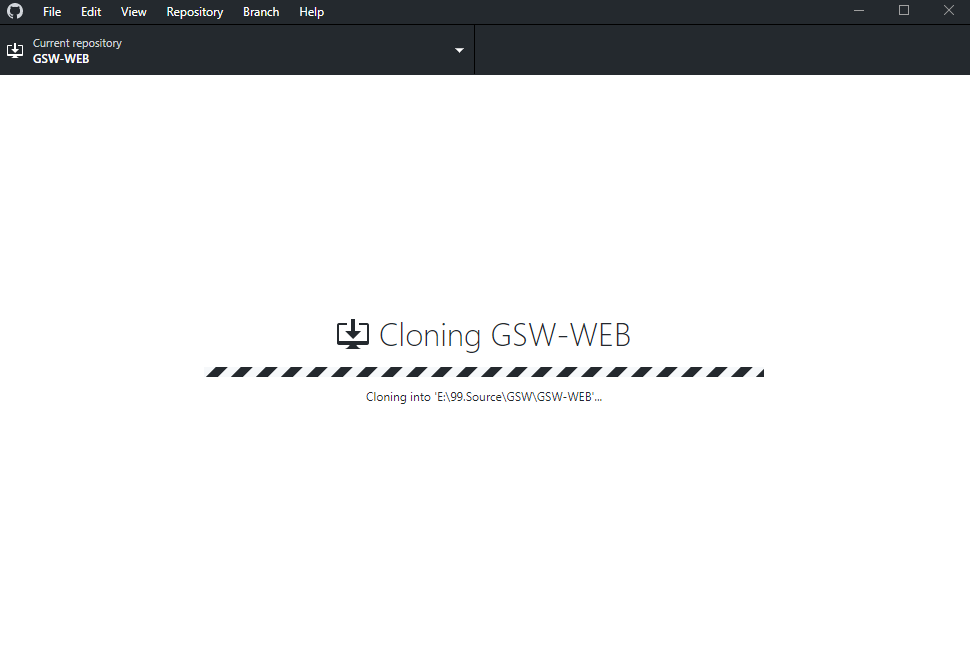
(2) Github Enterprise Server 탭에서 Github Desktop계정과 연동된 Repository리스트를 확인하고

“SGH-ISD/GSW-WEB”(그룹웨어&전자결재 저장소) 선택합니다.

Local Path에는 본인 PC와 Github Repository를 연결 할 경로를 설정 후 Clone을 클릭합니다.



아래와 같은 화면이 실행되며 Github에서 Local Path로 파일 가져오기 작업이 진행 됩니다.



**(3) 지정한 Local 경로에 폴더 및 파일들이 생성 되었는지 확인 합니다.**

1. ****브랜치(Branch)****

**개발을 진행 할 때 개발자들은 동일한 소스코드를 함께 공유하게 됩니다.**

**동일한 소스코드에서 어떤 개발자는 오류를 수정하고 또 다른 개발자는 새로운 기능을 개발할**

**수도 있습니다. 이와 같이 여러 사람이 동일한 소스코드를 기반으로 서로 다른 작업을 할 때**

**각각 서로 다른 버전의 소스코드가 만들어져 문제가 발생하는 경우가 있습니다.**

**이럴 때 여러 개발자들이 동시에 다양한 개발을 할 수 있게 만들어 주는 기능이 바로 브랜치(Branch) 입니다. 개발자가 브랜치를 생성하여 각자 독립된 작업영역(로컬 저장소)에서 마음대로**

**소스코드를 변경할 수 있으며 다른 브랜치와 병합(Merge)함으로써 작업한 내용들을 새로운**

**하나의 브랜치로 모을 수 있습니다.**

**A. 브랜치(Branch) 생성하기**

(1) Current branch에서 Master를 선택 하면 로컬 저장소에 Master의 소스를 가져 옵니다.

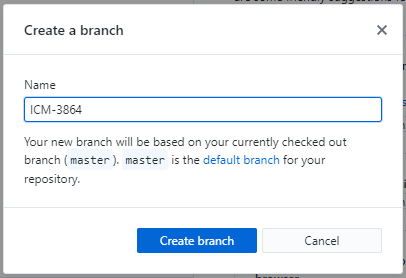
최근 배포된 소스가 있을 수 있으므로 “Fetch origin” 클릭하여 소스를 최신화 시켜줍니다.

(동일한 방법으로 다른 브랜치를 선택하면 해당 브랜치의 소스를 가져올 수 있음)

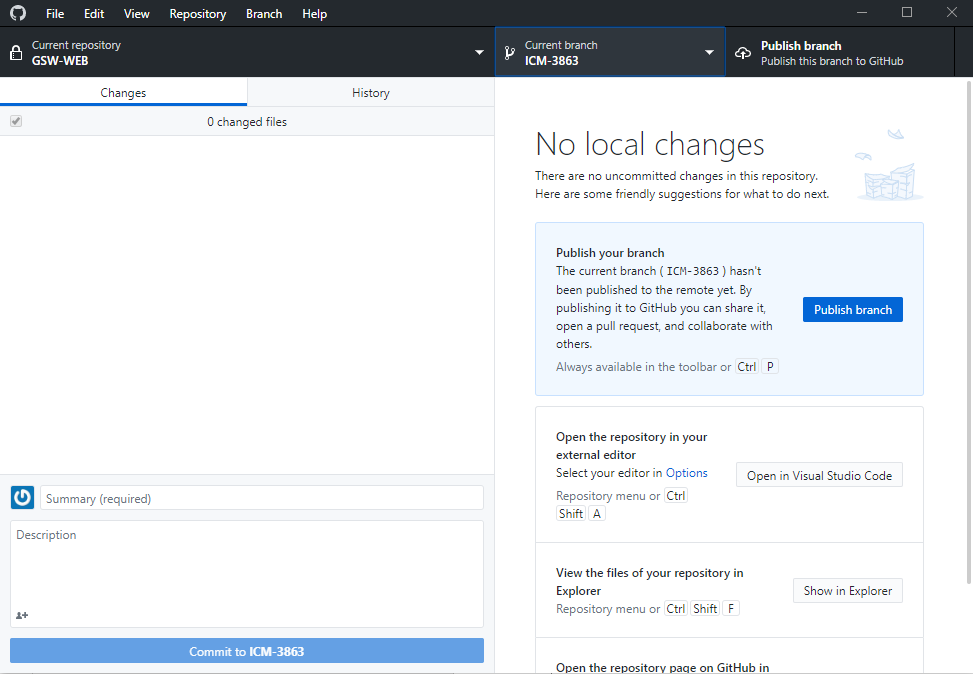
시스템 생성 대체 텍스트:
OFlIe Edit View 
Curren t repository 
GSW-WEB 
Repository 
Branch 
Help 
C」 branch 
master 
Branches 
Default branch 
Recent branches 
staging 
Other branches 
origin/3X:hEEHÆEl 2019-09-02 
2019-09-11 
origin/3X:hEEHÆEl 2019-09-23 
origin/3X*EEH시-Kg구족 
origin/CompanyLifeGuide 
origin,'℃M-3392 
Fetch origin 
Last fetched 26 minutes 凹0 
Pun r든국니ests 
New branch 
국 CIEYS 弓§0 
an hour 弓§0 
32 minutes 弓§0 
tudic Code 
8 ma-Iths 弓§0 
7 ma-Iths 弓§0 
7 ma-Iths 弓§0 
in Explorer 
8 ma-Iths 弓§0 
5 ma-Iths 弓§0 
an hour 弓§0 
00 듸tHub 
An updated ver듸이1 듸tHub 0E5kt09 is available and will be installed at the ne 
[h311§든드 
0 changed files 
Summary (req」ired) 
Description 
Com to master 
H드t이1로 
Choose 3 branch to merge into master 

(2) Github Desktop의 상단 메뉴에서 Branch > New branch 순으로 선택 한 후 Jira에서 배정 받은 업무의 티켓 번호 또는 업무명을 포함한 Name으로 Branch를 생성합니다.

(Current branch메뉴에서 분할할 Branch를 선택 후 Filter 우측에 “New branch” 버튼을 클릭하여도 Branch가 생성 됩니다.)



(3) 생성한 이름의 브랜치(Branch)가 생성되고 Current branch로 지정되어 있는지 확인 합니다.



1. ****커밋(Commi) 하기****

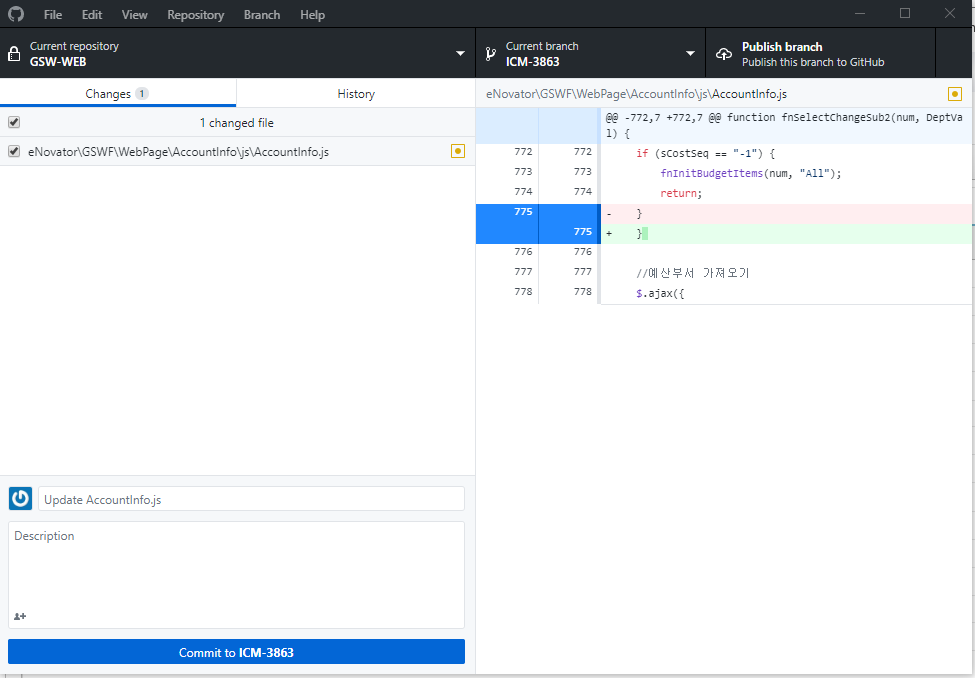
개발자가 업무에 의해 파일을 생성하거나 수정한다면 그 기록이 상단 Current branch에 설정된 브랜치(Branch)에 남게 되는데 이때 이 기록을 저장하는 것을 커밋(Commit)이라고 합니다.

(1) 파일 내용 수정 후 Commit 하기 전 확인사항

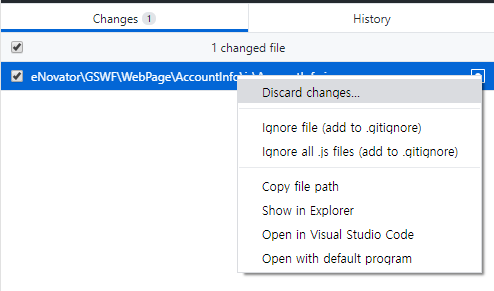
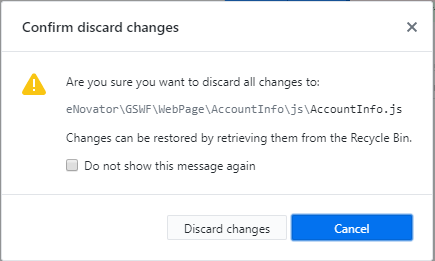
a. Current Branch가 내가 생성한 Branch가 맞는지 확인합니다.

b. Changes의 파일 리스트에 내가 수정한 파일만 존재하는지 확인합니다.

(소스코드를 빌드 하여도 변경 파일로 인지하니 Changes에 파일을 꼭 확인 해야함)

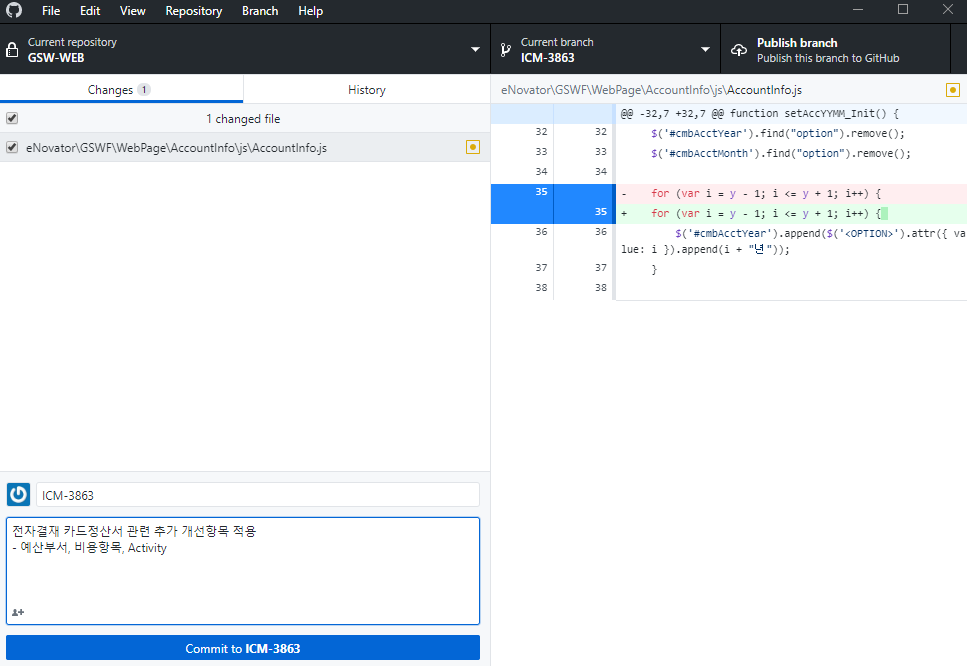


(2) Changed file에 불필요한 파일이 존재 시 우 클릭 후 “Discard Changes..” 선택하여 제거 합니다

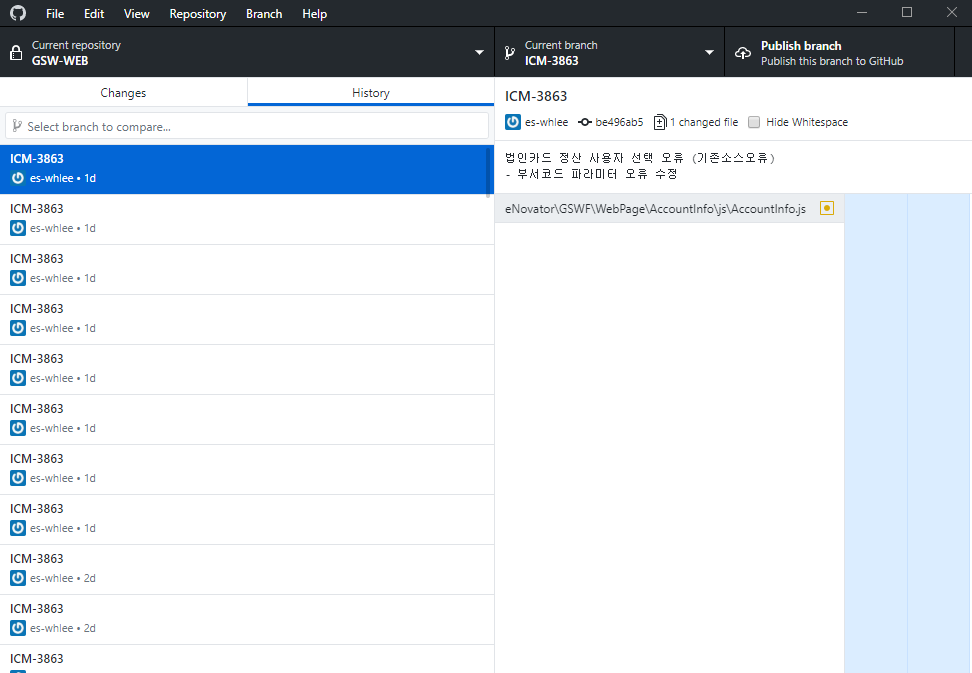
 

(2) Summary에 작업에 대한 제목(티켓 번호등..)을 기입하고 Description에 설명을 기입한 후

아래 “Commit to (선택된 브랜치명)” 버튼을 눌러 Commit 합니다.



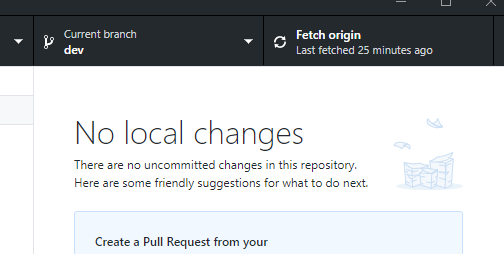
(3) History에서 선택된 브랜치(Branch)의 Commit 내역을 확인 할 수 있습니다.



1. ****병합(Merge) 하기****

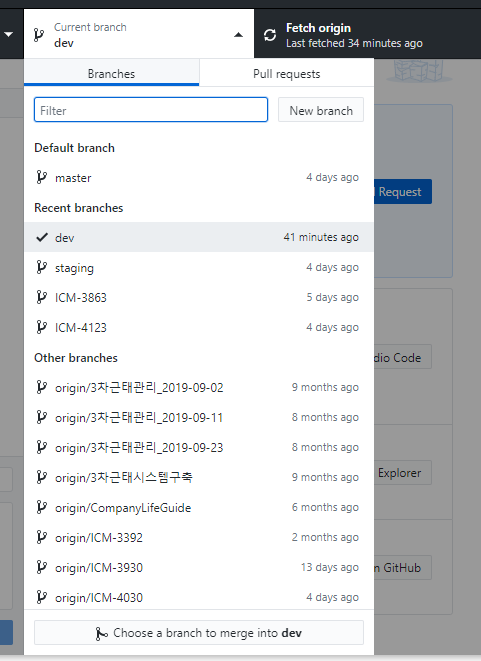
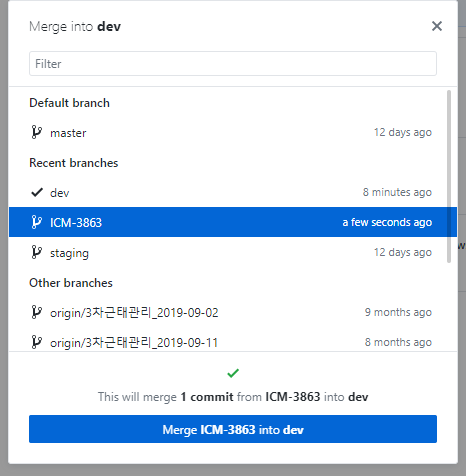
여러 사람들이 각각 생성한 브랜치(Branch)들을 원격 저장소에 저장하기 전 하나로 합치는 작업을 병합(Merge) 이라고 합니다.

(1) Current banch에서 “Dev” 브랜치(Branch)를 선택하고 “Fetch origin” 클릭하여 로컬 저장소로 가져온 소스를 최신화 시켜줍니다. (이 작업을 누락 할 경우 다른 사람의 소스와 꼬일 수 있음)

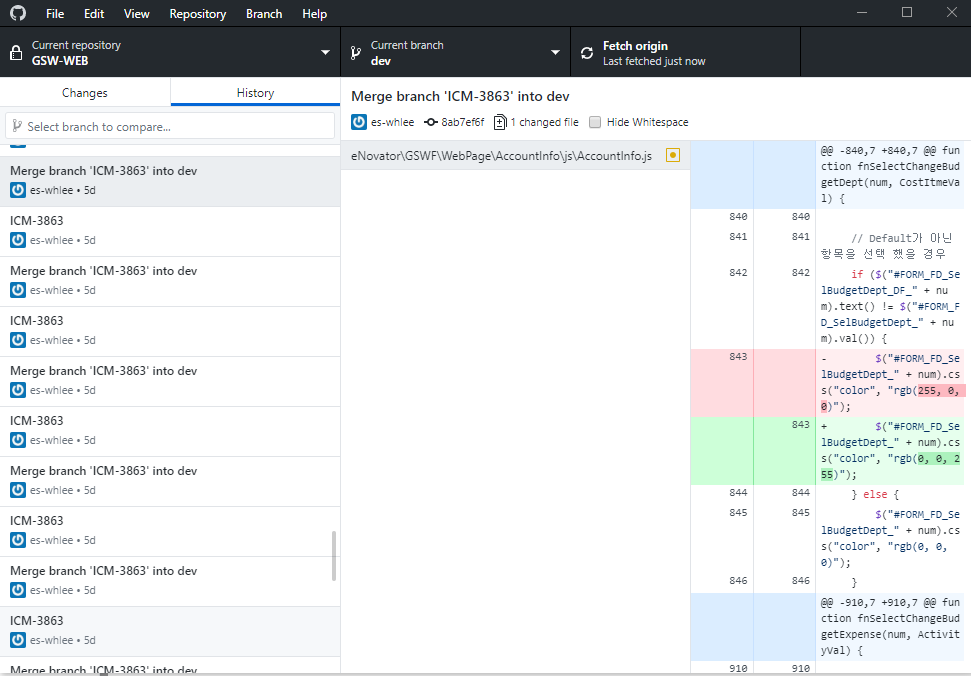


(2) Current branch에서 “Choose a branch to merge into (선택된 브랜치)”를 클릭하여 발생된

팝업창에서 커밋(Commit)한 이력이 있는 브랜치(Branch)를 선택하여 “Dev” 브랜치(Branch)와 파일들을 병합(merge) 할 수 있습니다.

(3) Dev” 브랜치(Branch)의 병합(merge) 이력들은 history에서 확인 할 수 있습니다.



1. ****푸시(Push) 하기****

병합(merge)하는 과정까지 로컬 저장소에서 진행되며, 원격 저장소에 업데이트하는

푸시(Push)를 해야 합니다.

(1) 병합(merge)까지 진행 후 Dev” 브랜치(Branch) 추가되는 history의 표시 이력들을

“Push origin”를 클릭하여 원격 저장소에 업데이트 합니다.

