

How to set git commit message template (feat. git hook)

🕒 Created	@Jul 20, 2020 3:08 PM
🏷️ Tags	Git

▼ 잡설

팀 프로젝트에서 커밋 메시지 템플릿을 사용하려고! 했는데! 우리는 GitLab을 쓰는 데!

아무리 검색을 해도 GitLab에서 설정하기~ 는 없는 거다!

그래서 Git Hook을 보면서.. 어 이거 어떻게 잘 만지면 될 거 같은데? 라고 생각했으 나

결론적으로, Github에서만 가능한 게 아니라 그냥 git이면 다 동일하게 설정하는 것이었다고 한다...

각설하고! 삽질하며 얻은 조금의 지식을 기록해보도록 하겠다.

Git Hook

내가 이해한 바에 따르면, git hook이란 다음과 같다.

| 특정 커맨드를 실행하기 전/후에 지정된 스크립트를 실행해주는 것

위의 특정 커맨드는 `./git/hooks` 에 저장된 sample 파일들을 보면 알 수 있는데, 뒤의 `.sample` 을 빼고 저장하면 커맨드 실행 시 해당 파일이 자동으로 실행된다.

나는 커밋 메시지의 템플릿을 바꾸고 싶었기 때문에 `commit-msg.sample` 을 열었는데...

```
# Doing this in a hook is a bad idea in general, but the prepare-commit-msg  
# hook is more suited to it.
```

아오케이... 이제 `prepare-commit-msg.sample` 을 열어보면 굉장히 영어가 많은데, 간단히

| commit message file을 수정하기 위해 사용되는 hook

정도의 의미라고 이해했다.

예시가 세 가지 정도 있는데, 각각의 경우 나오는 커밋 메시지는 다음과 같다.

1. The first one removes the "# Please enter the commit message..." help message.

```
# On branch master
# Your branch is up to date with 'origin/master'.
#
# Changes to be committed:
#   new file:   readme.md
#
# Untracked files:
#   commit_template.txt
#
```

써있는 그대로 Please enter... 을 지워준다.

2. The second includes the output of "git diff --name-status -r" into the message, just before the "git status" output. It is commented because it doesn't cope with --amend or with squashed commits.

```
A      readme.md
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Your branch is up to date with 'origin/master'.
#
# Changes to be committed:
#   new file:   readme.md
#
# Untracked files:
#   commit_template.txt
#
```

이 커밋에 포함된 파일 이름들을 커밋 메시지 description에 자동으로 추가해주는 듯?

3. The third example adds a Signed-off-by line to the message, that can still be edited. This is rarely a good idea.

```
Signed-off-by: Hyejin Park <triaina0@gmail.com>
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Your branch is up to date with 'origin/master'.
#
# Changes to be committed:
#   new file:   readme.md
#
# Untracked files:
#   commit_template.txt
#
```

작성자 이름을 커밋 메시지 description에 자동으로 추가해준다

인터넷에 검색해보면 이 파일을 잘 수정해서 최근 commit log를 나오게 한다든가 하는, 좋은 레퍼런스들이 많은데...

일단 셸 스크립트(가 맞나?)를 잘 모르는 나로서는 작동 구조를 도저히 파악할 수가 없었다.

그러다 "잠깐, `commit.template`이라는 명령어가 github에서만 사용 가능하다고...?" 하는 생각이 들었다...! 그렇다 이제야 깨달은 것이다...

commit template

commit template은 정말 간단하게 설정 가능한데,

1. 커밋 템플릿으로 사용할 파일 생성하기

나는 이분의 내용을 그대로~ 까진 아니고, 조금 수정해서 사용했다.

▼ 이렇게!

[예약어] 지라 이슈 번호 : 제목(한 일)

```
# 제목의 길이는 최대 40글자까지 한글로 간단 명료하게 작성
# 제목을 작성하고 반드시 빈 줄 한 줄을 만들어야 함
# 제목에 . (마침표) 금지
# --- 예약어 리스트 ---
# feature : 새로운 기능
# bug      : 버그 수정
# update   : 비즈니스 로직 변경
# docs     : 문서 (문서 추가, 수정, 삭제)
# test     : 테스트
# etc      : 기타 변경사항
```

세부 내용 기술

```
# 내용의 길이는 한 줄당 60글자 내외에서 줄 바꿈. 한글로 간단 명료하게 작성
# 어떻게 보다는 무엇을, 왜 변경했는지를 작성할 것 (필수)
```

원하는 장소에 아무 이름으로 만들어주면 되는데, 나는 깃 프로젝트 폴더 루트에 `.gitmessage` 라는 이름으로 만들었다. ~~지금 보니까 이름 이상하네;~~

2. 설정 명령어 입력하기

Linux/MacOS라면 터미널에 바로 입력해주면 될 것이고, Windows라면 git bash를 켜서 입력해주자.

```
git config [--global] commit.template [파일 경로]
```

~~이거 그대로 복사하면 아무것도 안됩니다.. 다들 아시겠지만! 혹시나 해서!~~

`--global` 옵션은 이 파일을 모든 로컬 리포지토리에서 사용할 때 사용하면 된다.

파일 경로는 절대 경로를 입력해줘도 되고 상대 경로를 입력해줘도 된다.

끝이다 ^. ^

다른 사람 레포에도 템플릿 설정하기

이 부분에서 고민을 했었는데, 간단히 Shell script를 만들어서 모두가 실행해달라고 하기로 했다.

우선 shell script 파일을 만들고, 이렇게 입력해주었다. `파일 이름.sh` 형태로 만들어주면 된다.

```
#!/bin/bash

git config commit.template [파일 경로]
```

여기서는 `--global` 을 쓰면 안되겠죠? 이 레포에서만 쓸 템플릿이니까 ^. ^

파일 경로는 `./` 을 기준으로 상대 경로를 입력해주자.

Windows같은 경우 스크립트를 더블 클릭만 하면 실행된다.

~~Mac은... 키게 되면 쓰겠다~~

결과

```

[예약어] 지라 이슈 번호 : 제목 (한 줄)

# 제목의 길이는 최대 40글자까지 한글로 간단 명료하게 작성
# 제목을 작성하고 반드시 빈 줄 한 줄을 만들어야 함
# 제목에 .(마침표) 금지
# --- 예약어 리스트 ---
# feature : 새로운 기능
# bug      : 버그 수정
# update   : 비즈니스 로직 변경
# docs     : 문서 (문서 추가, 수정, 삭제)
# test     : 테스트
# etc      : 기타 변경 사항

세부 내용 기술

# 내용의 길이는 한 줄당 60글자 내에서 줄 바꿈. 한글로 간단 명료하게 작성
# 어떻게 보다는 무엇을, 왜 변경했는지를 작성할 것 (필수)
# On branch master
# Your branch is up to date with 'origin/master'.
#
# Changes to be committed:
#   new file:   hi
#

```

이런 식으로 나오게 된다.