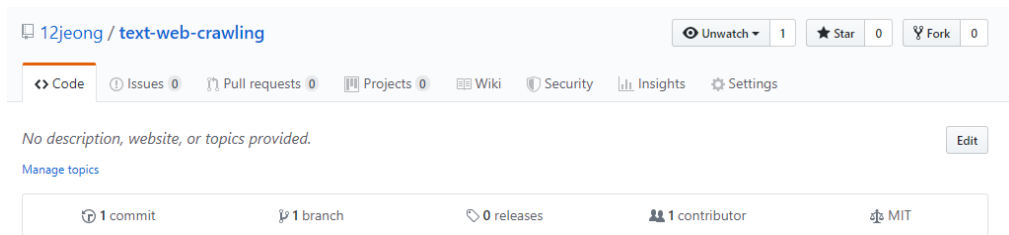


1. 메인유저는 함께 참여할 프로젝트의 repository를 만듭니다. (Public 으로 만들 것!)

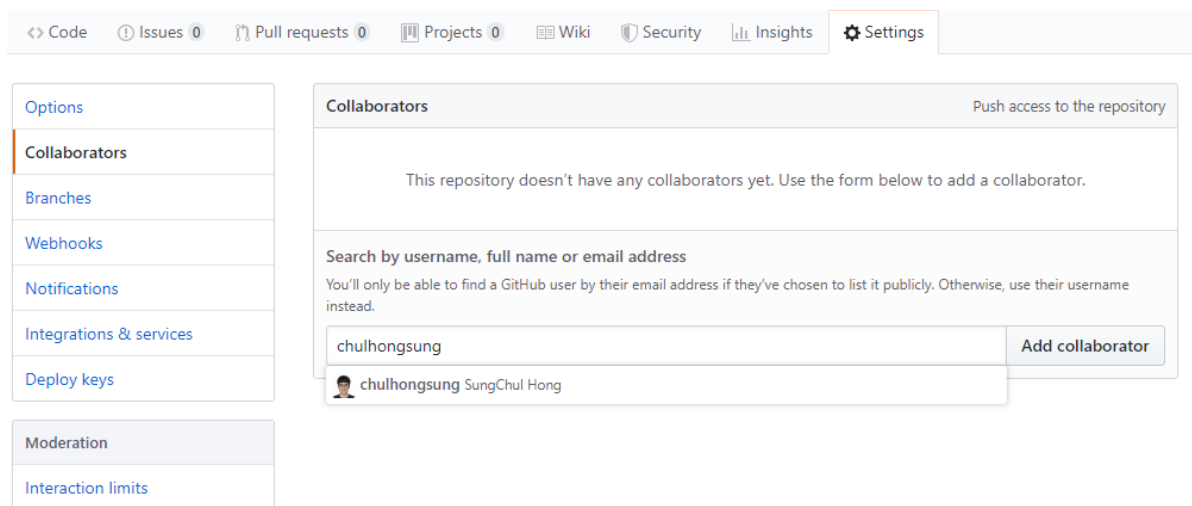
예시) text-web-crawling 을 만들었습니다.



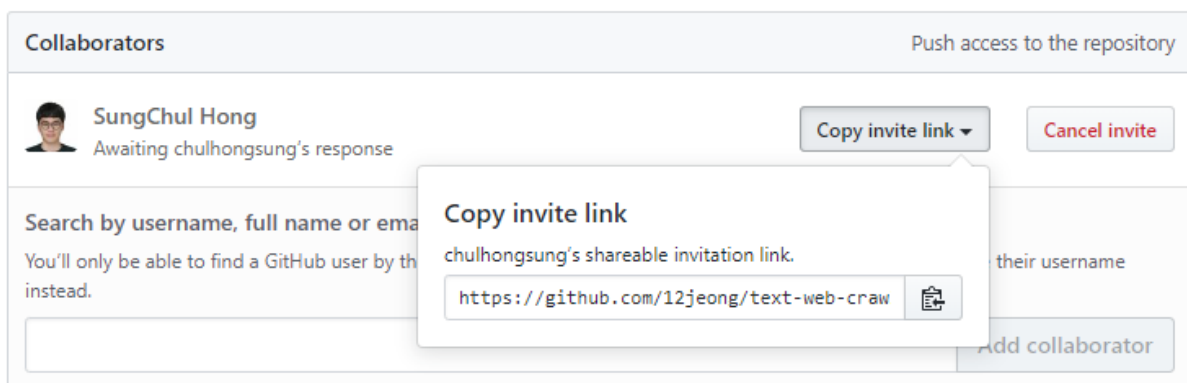
2. 메인유저는 우선 github desktop을 통해 로컬저장소를 만들고 파일을 업로드할 수 있습니다.

(09/20/2018, IJeong Han, For Github Desktop beginner 참고)

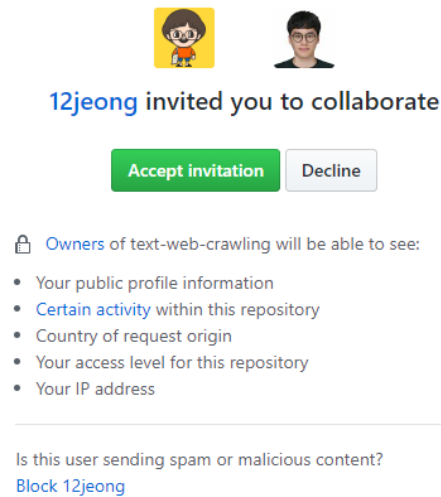
3. 메인유저는 협업할 사람을 초대합니다. (Settings > Collaborators > Add collaborator)



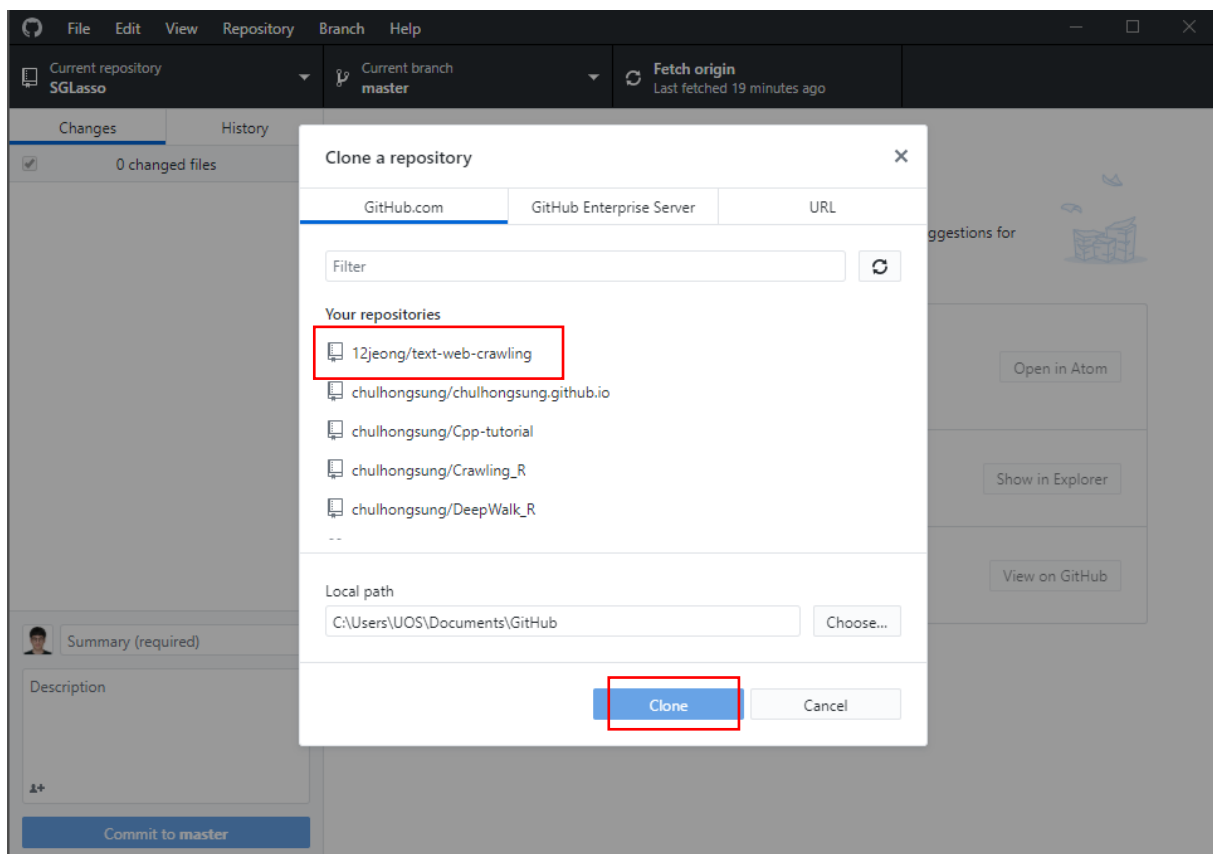
초대받은 사람에게 메일이 도착합니다. 또는 아래의 초대 링크를 메시지로 보내도 됩니다.



4. Accept를 받으면 함께 협업할 수 있습니다!

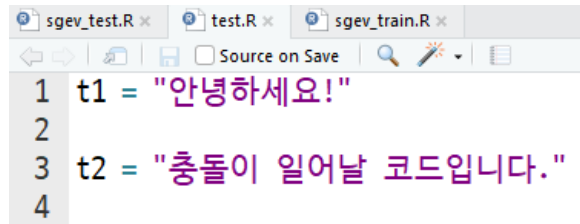


5. 초대받은 사람도 이제 로컬 저장소를 만들 수 있습니다.



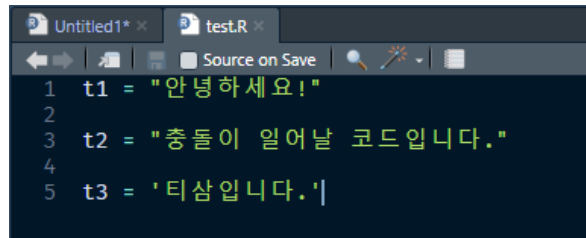
6. 실험1

6-1) repository에 올려져있던 코드입니다.



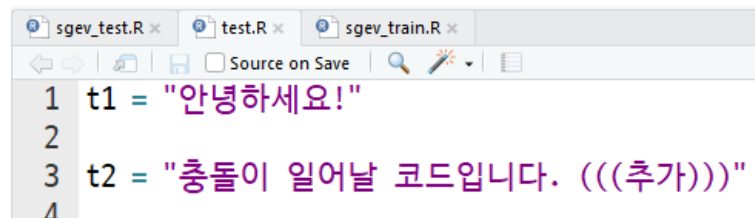
```
sgev_test.R x test.R x sgev_train.R x
Source on Save
1 t1 = "안녕하세요!"
2
3 t2 = "충돌이 일어날 코드입니다."
4
```

6-2) 유저1이 코드를 수정/ commit, push 했습니다.



```
Untitled1* x test.R x
Source on Save
1 t1 = "안녕하세요!"
2
3 t2 = "충돌이 일어날 코드입니다."
4
5 t3 = '티삼입니다.'
```

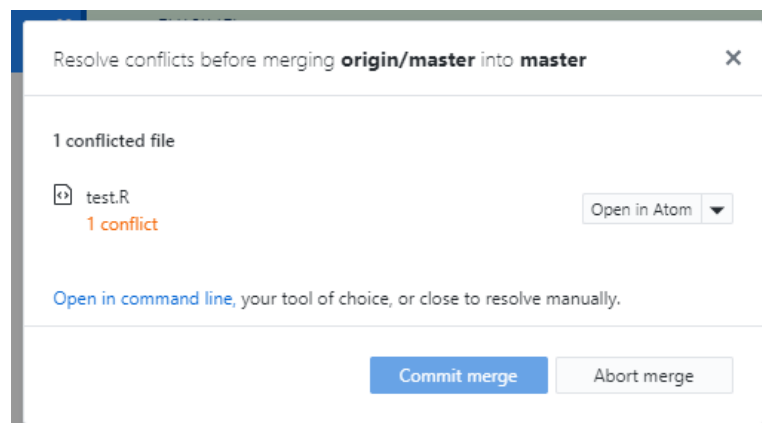
6-3) 동시간대 유저2가 코드를 수정하다가 commit, push 합니다. 그러나,



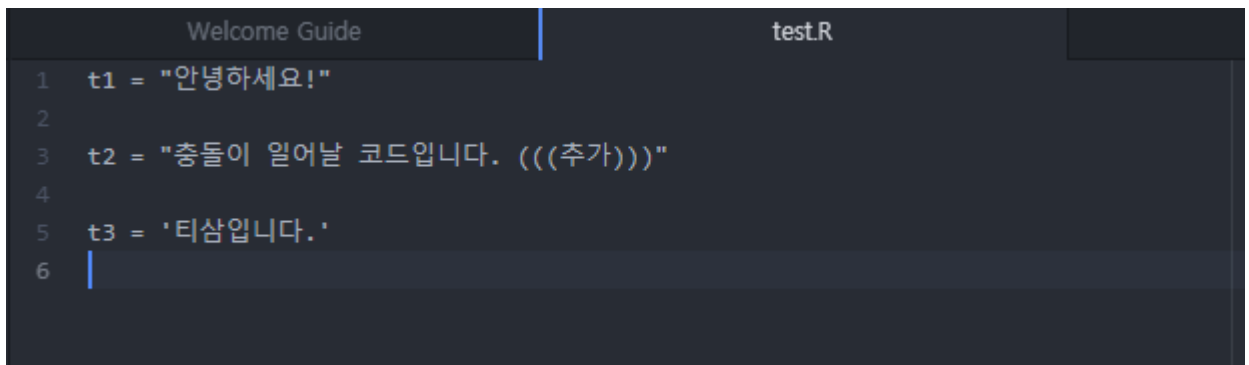
```
sgev_test.R x test.R x sgev_train.R x
Source on Save
1 t1 = "안녕하세요!"
2
3 t2 = "충돌이 일어날 코드입니다. (((추가)))"
4
```

6-4) 유저2에게 conflict 메시지가 출력됩니다. Editor 프로그램을 이용해 Conflict를 해결해야합니다.

(Open in Atom / (또는 다른 프로그램) 버튼을 누릅니다)

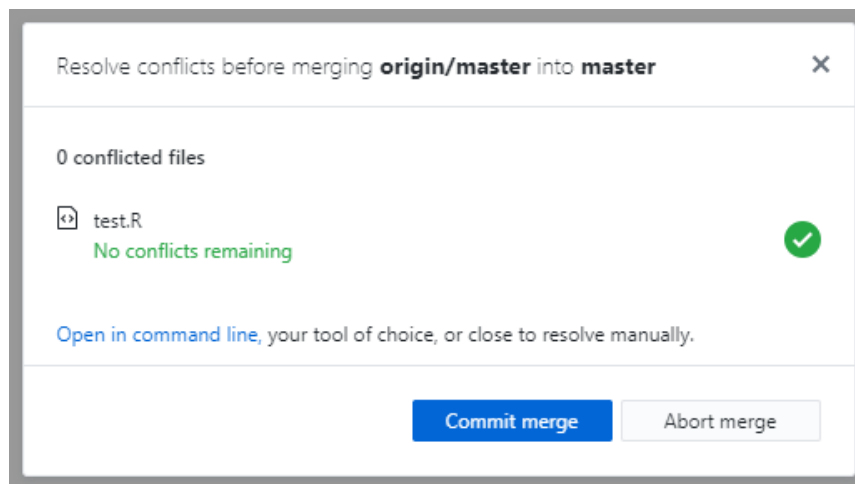


6-5) Editor를 통해 최종으로 사용할 코드로 수정해줍니다.

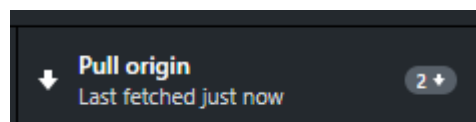


```
1 t1 = "안녕하세요!"
2
3 t2 = "충돌이 일어날 코드입니다. (((추가)))"
4
5 t3 = '티삼입니다.'
```

6-6) 이제 유저2는 push를 할 수 있습니다 ☺ (Commit merge > Push)

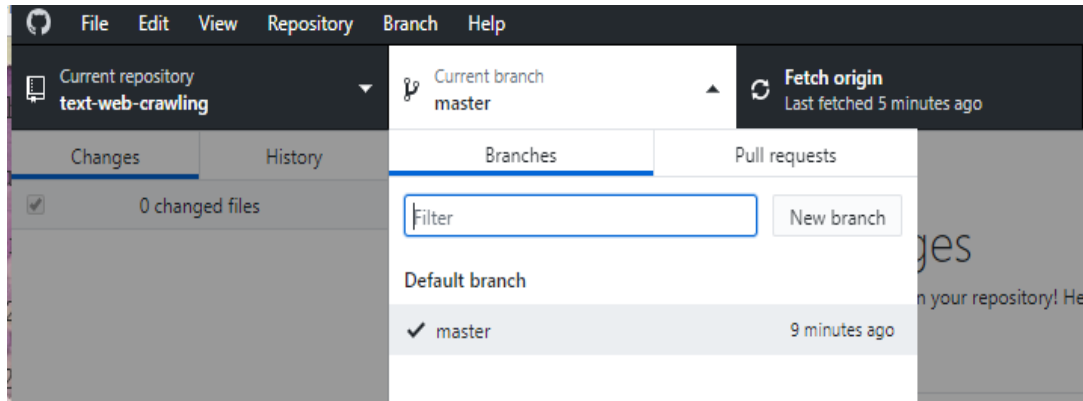


6-7) 유저1은 수정된 코드를 pull을 통해 로컬저장소로 업데이트 할 수 있습니다.



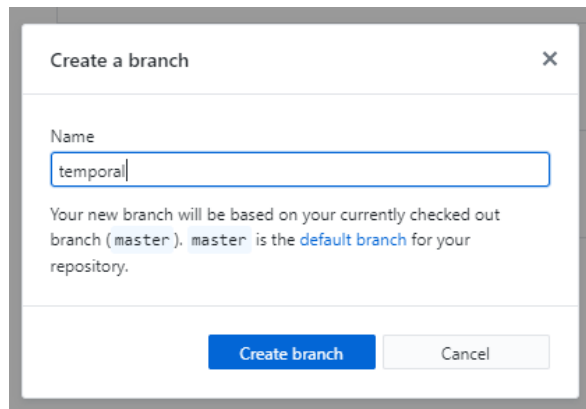
7. 실험2 (branch 사용법) : 협업을 위해 규칙을 정합니다.

규칙1) 'master' branch 에서는 완성된 코드만 저장합니다.

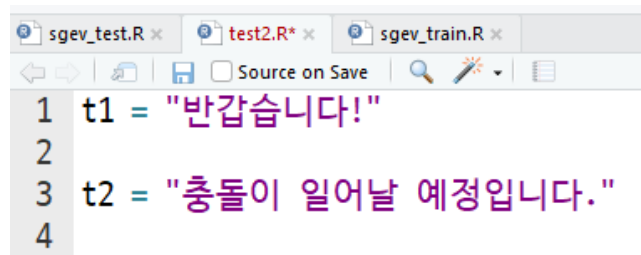


규칙2) 각자 작업하는 코드는 다른 branch 'temporal' 에 저장합니다.

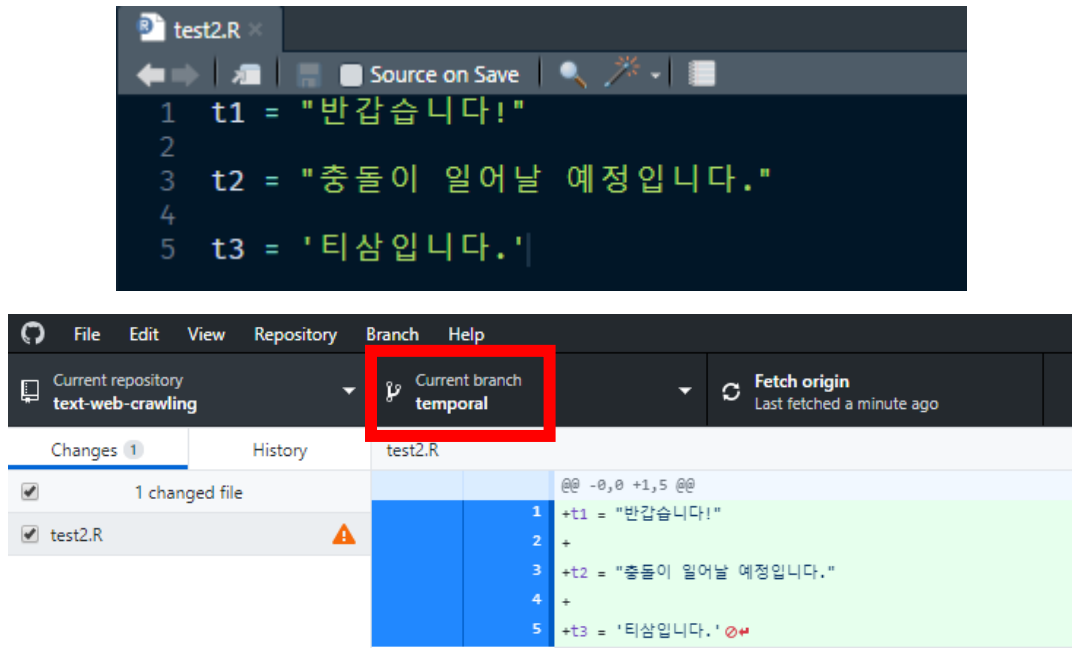
새로운 Branch 는 Branch > New branch 를 통해 만들 수 있습니다.



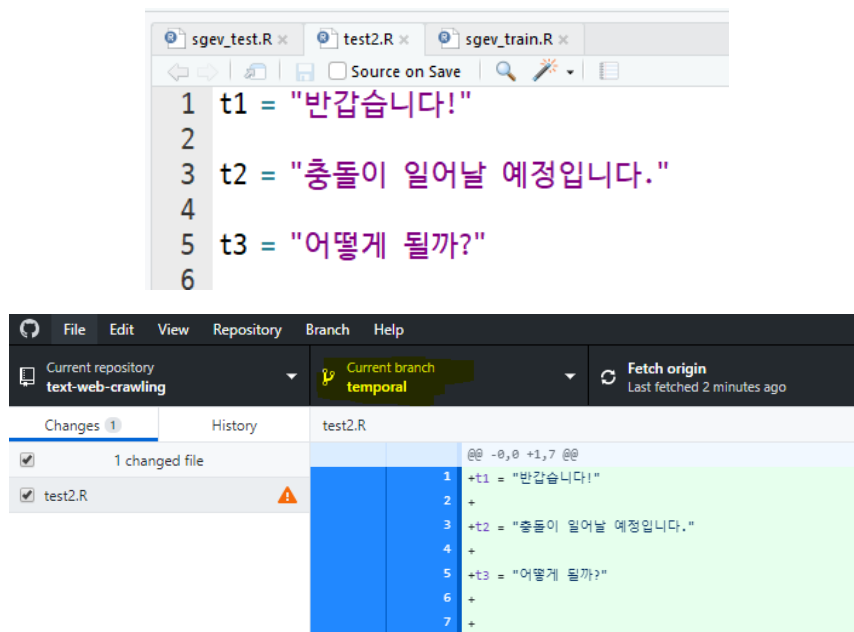
7-1) repository에 올려져 있던 코드입니다. (in 'master' branch)



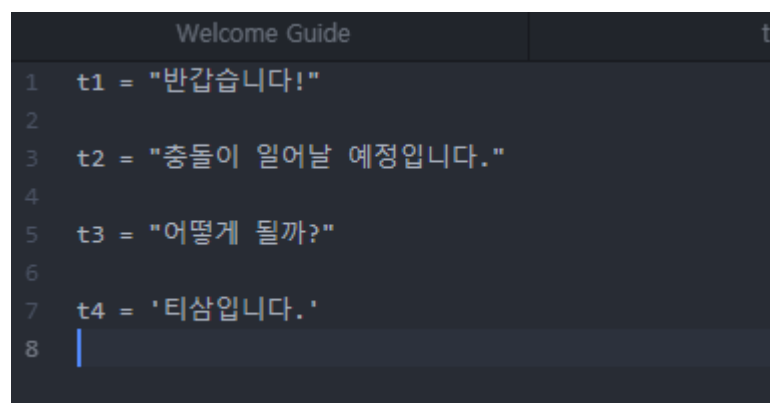
7-2) 유저1는 코드를 수정하여 'temporal' branch를 통해 push 합니다.



7-3) 동시간대 유저2도 같은 코드를 수정하고 'temporal' branch에 push 합니다.



7-4) 충돌이 생겨 atom을 통해 코드를 수정하였습니다. (6-4, 6-5 참고)



7-5) 다 완성된 코드를 master에 올리고자 합니다.

Github 웹에서 Pull requests > New pull request 를 누르고 'temporal'에서 'master'로 업데이트를 요청

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: master ← compare: temporal X Can't automatically merge. Don't worry, you can still create the pull request.

update plz master!

Write Preview

1) user1 added t3
2) user2 added t4

Attach files by dragging & dropping, selecting or pasting them.

Create pull request

7-6) master와의 충돌이 있다면 (Resolve conflicts) 해결하고 (Mark as resolved) Commit merge) 클릭!

Add more commits by pushing to the **temporal** branch on **12jeong/text-web-crawling**.

This branch has conflicts that must be resolved
Use the [web editor](#) or the [command line](#) to resolve conflicts.

Conflicting files
test2.R

Merge pull request

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

12jeong / text-web-crawling

<> Code ① Issues 0 Pull requests 1 Projects 0 Wiki Security Insights Settings

update plz master! #6

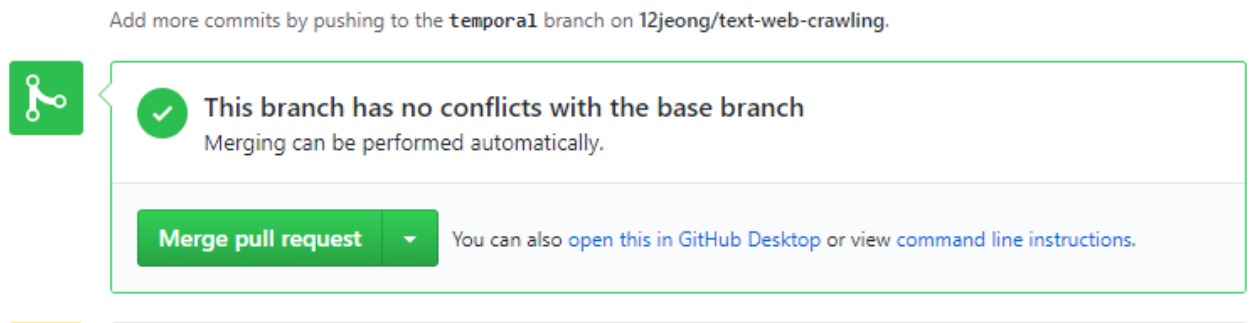
Resolving conflicts between **temporal** and **master** and committing changes to **temporal**

1 conflicting file

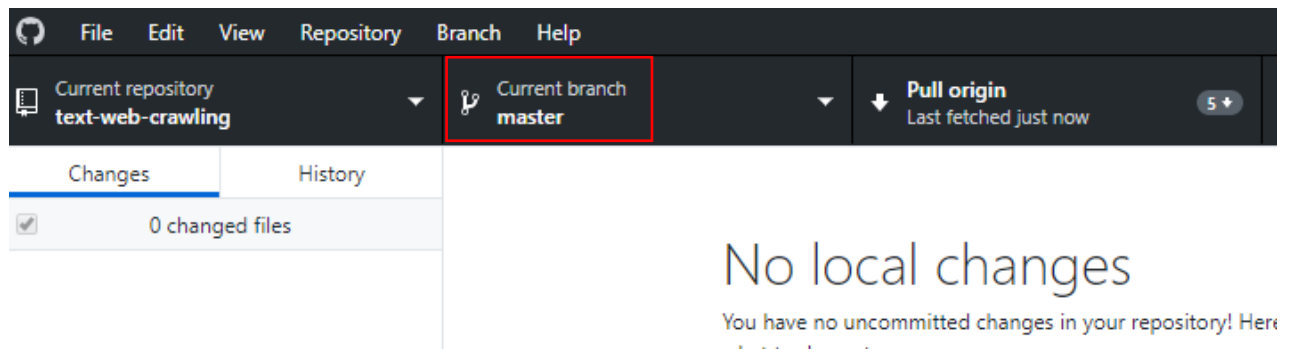
test2.R

```
1 t1 = "반갑습니다!"
2
3 t2 = "충돌이 일어날 예정입니다."
4
5 <<<<<< temporal
6 t3 = "어떻게 될까?"
7
8 t4 = '티샵입니다.'
9 =====
10 >>>>>> master
11
```

7-7) 충돌이 없어졌다면 업데이트 (Merge pull request) Commit merge)



7-8) 사용자들은 최종 코드를 'master' branch에서 받아올 수 있습니다. (Pull)



감사합니다 😊

도움을 주신 chulhongsung님 감사합니다.