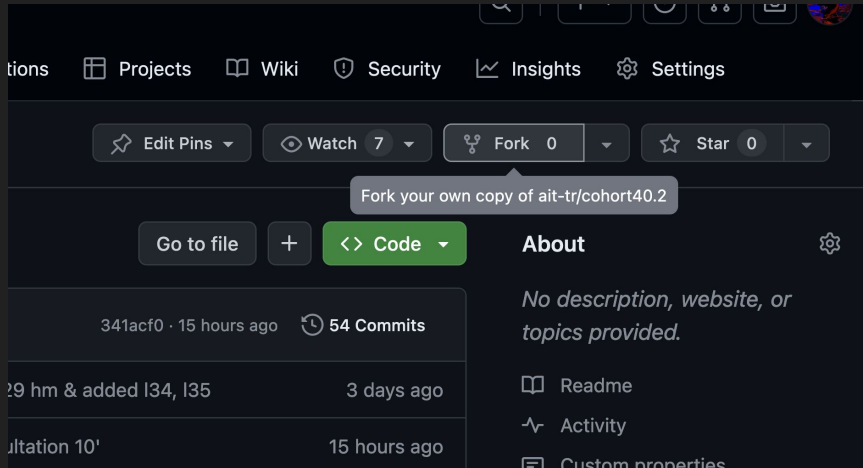




# Git branches



# Branching



Ветвление в Git позволяет разработчикам изолированно работать над различными функциями, исправлениями или экспериментами без влияния на основную кодовую базу.

Это обеспечивает безопасное развитие проекта и упрощает совместную работу.

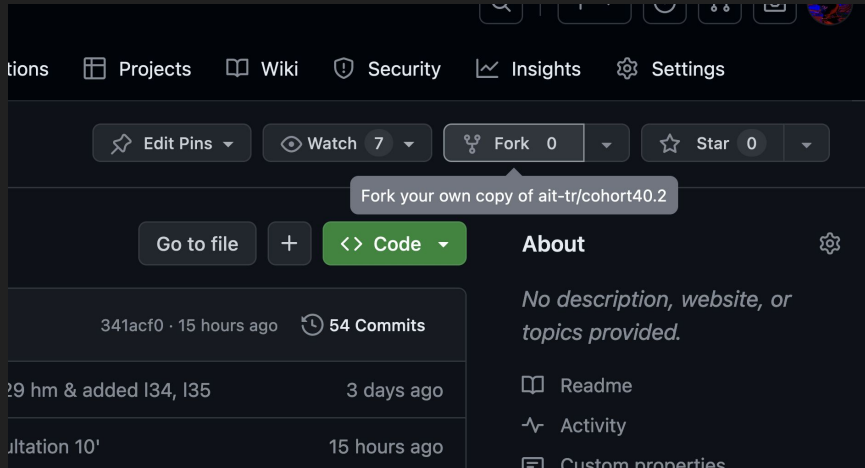


# Fork

Fork в Github — это создание своей копии в качестве удаленного репозитория.

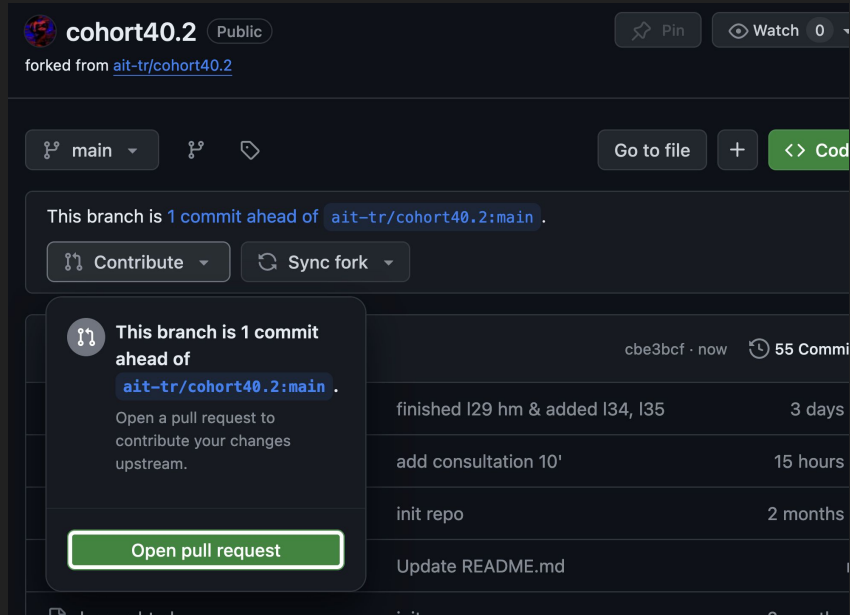
Fork необходим для того, чтобы вы имели возможность работать с удаленным репозиторием (пусть и с его копией) в тех случаях, когда у вас нет права редактировать файлы в основном репозитории.

Документация от github по [ссылке](#)  
Сценарий использования по [ссылке](#)  
Сценарий использования попроще по [ссылке](#)





# Pull request



Pull request — это запрос на включение ваших изменений из одной ветки - в другую, в том числе это возможно делать из между репозиторием и его fork'ом.

Вы работаете над какими то изменениями на своей отдельной версии или в своей отдельной копии основго репозитория (в форме основного репозитория), когда вы хотите добавить свои изменения в основную ветку, вы должны открыть pull request.

Сценарий использования по [ссылке](#)

Сценарий использования попроче по [ссылке](#)



# Create a branch

```
→ myRepo git:(master) git branch new_branch
→ myRepo git:(master) git branch new_branch
fatal: a branch named 'new_branch' already exists
→ myRepo git:(master) git checkout -b new_branch_2
Switched to a new branch 'new_branch_2'
```

Для создания ветки используется команда **git branch**

Она создает новую ветку на основе вашего текущего состояния: новая ветка будет обладать тем же состоянием (в т.ч. теми же незафиксированными файлами)

При попытке создать ветку с тем же названием возникнет ошибка

Чтобы создать ветку и сразу переключиться на нее можно использовать **git checkout -b new\_branch\_2**



# Checkout

```
→ ex2 git:(new_branch_2) git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
→ ex2 git:(main) git checkout unknown_branch
error: pathspec 'unknown_branch' did not match any
file(s) known to git
```

Для переключения на другую ветку используется команда **git checkout main**

Если ветки указанной не существует, то будет выведена ошибка

Если состояние другой ветки конфликтует с текущими незафиксированными изменениями, то возникнет ошибка

Если все пройдет успешно, то вы переключитесь на состояние другой ветки, незафиксированные измененные файлы, которые не конфликтуют с версией другой ветки так же останутся доступны



# Delete

```
→ ex2 git:(main) git branch -d new_branch  
Deleted branch new_branch (was dd9072c).  
→ ex2 git:(main) git branch -d new_branch_2  
Deleted branch new_branch_2 (was dd9072c).  
→ ex2 git:(main) git branch -d new_branch_3  
error: branch 'new_branch_3' not found.
```

Для удаления ветки используется команда  
**git branch -d new\_branch**

Если ветки указанной не существует, то  
будет выведена ошибка

Вы не можете удалить ветку на которой  
находитесь

# Conflicts

```
→ myRepo git:(master) git branch new_branch
→ myRepo git:(master) echo somestring > Main.java
→ myRepo git:(master) X git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   Main.java

no changes added to commit (use "git add" and/or "git commit -a")
→ myRepo git:(master) X git add Main.java
→ myRepo git:(master) X git commit -m "somestring"
[master dba22e1] somestring
 1 file changed, 1 insertion(+), 112 deletions(-)
→ myRepo git:(master) git checkout new_branch
Switched to branch 'new_branch'
→ myRepo git:(new_branch) echo someNEWstring > Main.java
→ myRepo git:(new_branch) X git add Main.java
→ myRepo git:(new_branch) X git commit -m "someNEWstring"
[new_branch 360fab0] someNEWstring
 1 file changed, 1 insertion(+), 112 deletions(-)
→ myRepo git:(new_branch) git merge master
Auto-merging
basic_programming/lesson_29/code/examples/src/129/myRepo/Main.java
CONFLICT (content): Merge conflict in
basic_programming/lesson_29/code/examples/src/129/myRepo/Main.java
Automatic merge failed; fix conflicts and then commit the result.
→ myRepo git:(new_branch) X cat Main.java
<<<<<<< HEAD
someNEWstring
=====
somestring
>>>>>> master
```

Конфликты возникают, когда Git не может автоматически объединить изменения из разных веток.

Для их разрешения необходимо вручную изменить файлы, содержащие конфликты, и затем добавить их в индекс.

См. лекцию

Чтобы воссоздать конфликт в локальном репозитории и решить конфликт удобными инструментами:

- git checkout master
- git pull
- git checkout new\_branch
- git merge master





# Контрольная точка

- Понятно ли?

Если все ясно, ставим плюсы, иначе - задаем вопросы.



# Домашнее задание

Материалы:

- про ветвление подробно: [ссылка](#)
- коротко про github еще раз: [ссылка](#)



The end