

Практическое занятие № 5 — 6

Тема: Работа с удаленным репозиторием

Цель: Выработать навыки работы с удаленным репозиторием при индивидуальной и совместной работе на проекте.

Пояснения. Выполнить глобальные настройки git. Команды выполнять в консоли.

!!!Внимание!!! По окончании работы с удаленным репозиторием (в конце урока) необходимо удалить ssh-ключ на github, локальный файл с ключом не удалять.

Ход работы

1. В рабочей папке создать проект (например, в IDE PyCharm).
2. В папке с рабочим проектом выполнить:

git init

3. Убедиться в наличии файлов (особенно .git) и папок в проекте:

ls -a (-a показывает все файлы, в т.ч. скрытые)

4. Создать SSH – ключ

ssh-keygen -t rsa

В процессе генерации необходимо ввести:

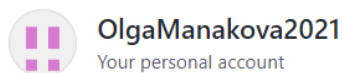
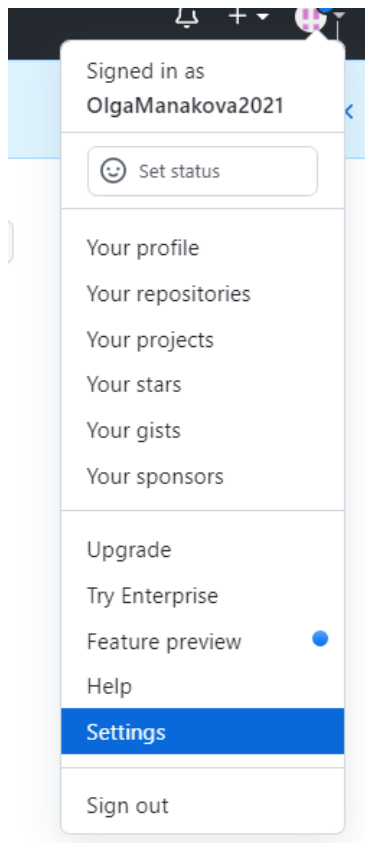
1. общее имя файла для частного и публичного ключей (например, <test>)
 2. для повышения защиты ключей ввести пароль, повторить его еще раз.
- Процедура не является обязательной, но желательной.

Два ключа появятся в корневой папке пользователя.

5. Зайти в файл публичного ключа:

cat <имя вашего ключа>.pub

Скопировать все содержимое для дальнейшего переноса в настройки аккаунта. Залогиниться в своем аккаунте на GitHub. Затем в настройках аккаунта выполнить следующие действия (см. рисунки):



- Public profile
- Account
- Appearance
- Accessibility
- Notifications

Access

- Billing and plans
- Emails
- Password and authentication
- Sessions
- SSH and GPG keys
- Organizations
- Moderation

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

SSH keys / Add new

Title

1

Key type

Authentication Key ↕

Key

2

3

6. Выполнить проверку подключения по SSH к сайту (в нашем случае это GitHub):

```
ssh -t git@github.com
```

После проверки будет два события:

- требование отпечатка ключа, написать yes
- вам будет отказано в доступе. Для предоставления доступа выполнить команду:

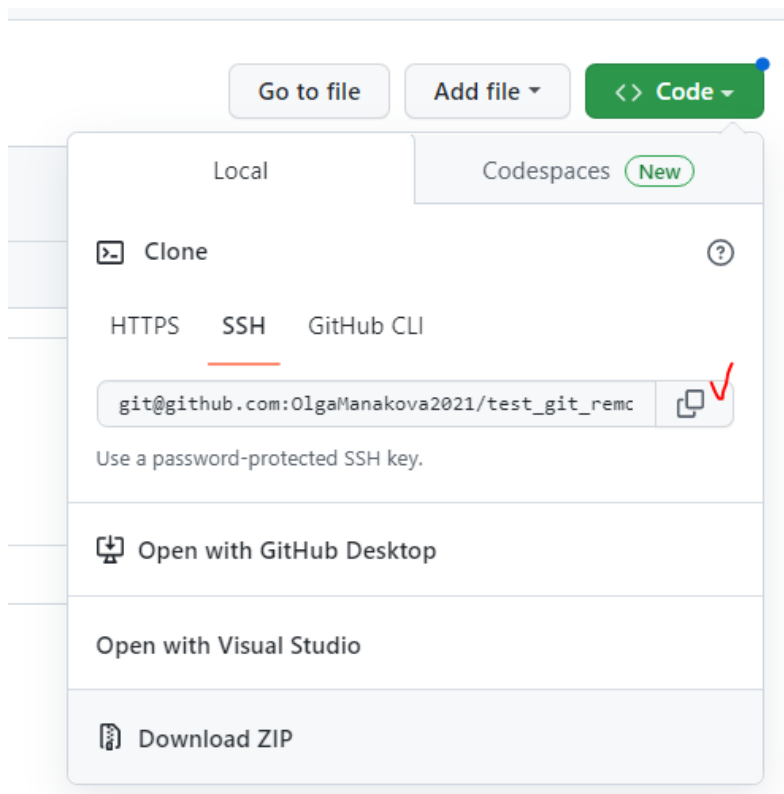
```
ssh-add <имя вашего ключа>
```

7. Создать репозиторий на GitHub (например, test_git_remote).

8. Подключиться к созданному репозиторию:

```
git remote add <имя подключения> <ssh-ссылка на репозиторий>
```

где <имя подключения> - рекомендуется писать origin (или любое имя), для получения <ssh-ссылка на репозиторий> выполнить:



9. Убедиться, что подключение состоялось

git remote -v
или
Git remote -verbose

10. По умолчанию в удаленном репозитории будет ветка main, в локальном – master. Ветки main нет в локальном репозитории, но ее можно добавить:

git fetch

Команда позволяет получить изменения из удаленного репозитория, при этом не записывая их в локальный репозиторий.

git fetch «подтянет» все ветки и файлы из удаленного репозитория. Убедимся в этом:

git switch main

git branch

11. Создадим текстовый файл на ветке main в удаленном репозитории на github.

12. Получим все актуальные изменения, ветки, папки, в т.ч. созданный файл в локальный репозиторий из удаленного:

git pull

13. Перенесем весь проект в удаленный репозиторий, т.е. выполним привычный набор команд:

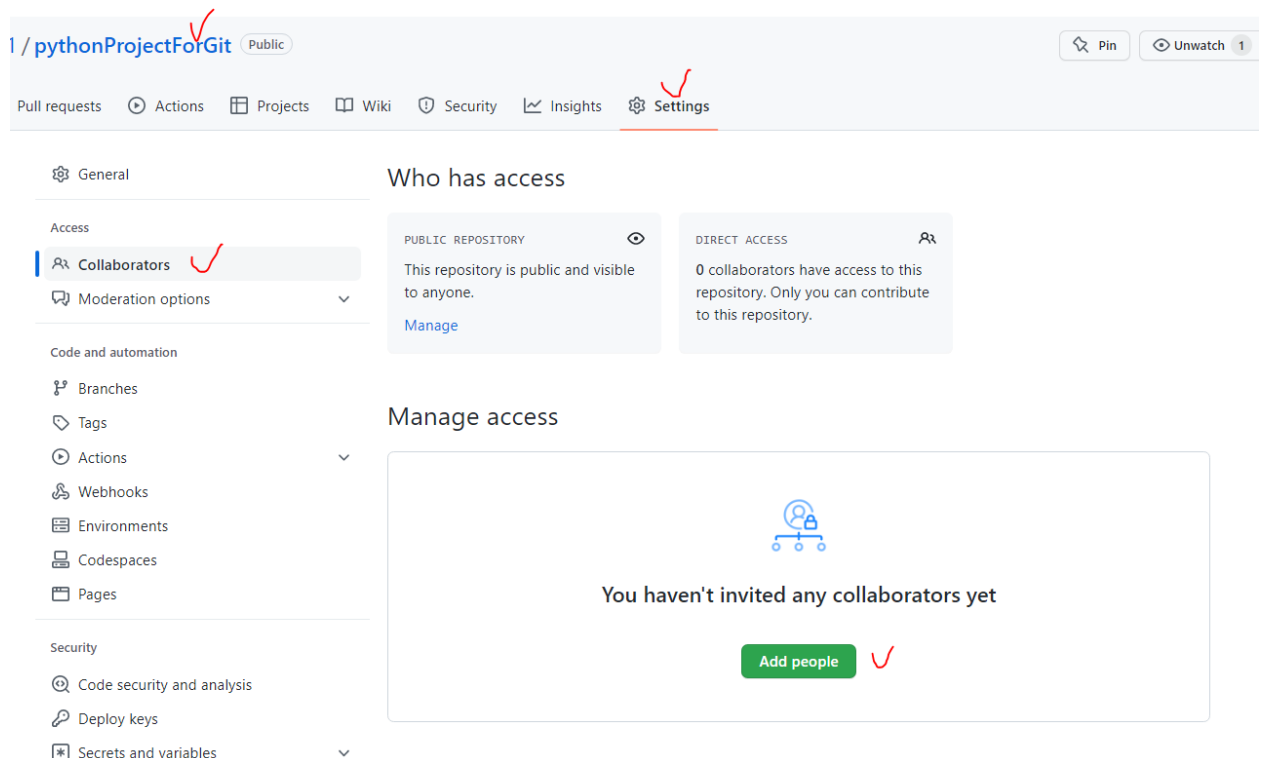
```
git add .  
git commit -m "..."  
git push
```

при выполнении `git push` может появиться ошибка. Чтобы ее исправить - выполнить копирование предложенной команды с параметрами. Затем выполнить ее.

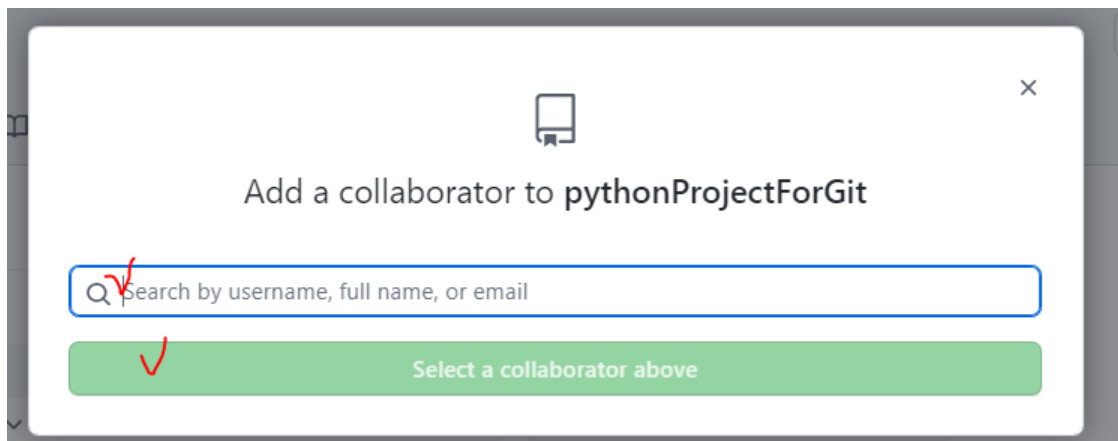
Совместная работа над проектом в git/github

Перед выполнением нижеприведенных действий убедиться, что все коммиты завершены

Пригласим второго участника проекта через github:
Зайти в репозиторий проекта
Выбрать Settings

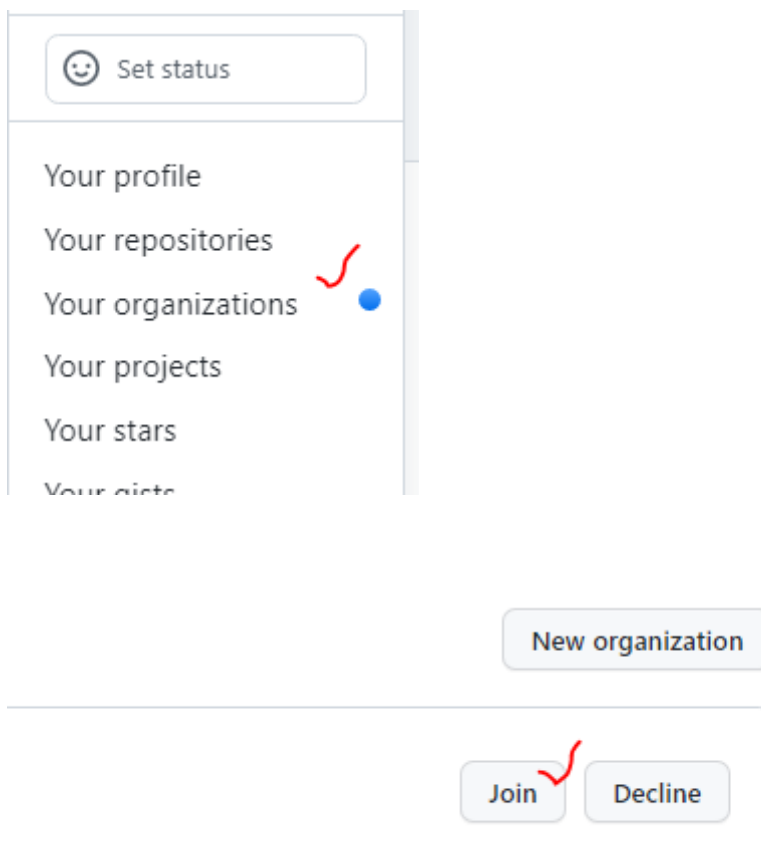


В появившемся окне



Ждем подтверждения приглашения от второго участника.

В github второго участника придет приглашение:



После нажатия на *Join* второй участник присоединится к проекту. Затем он должен выполнить клонирование данного проекта на свой ПК:

git clone <http-ссылка на репозиторий>

Владелец проекта создает новый файл (test.txt) и пушит его в удаленный репозиторий (команды подаются через git).
Второй участник выполняет

git pull

и убеждается (команда `ls`), что файл `test.txt` попал в его локальный репозиторий.

Второй участник проекта создает новый файл (`test1.txt`) и пушит его в удаленный репозиторий (команды подаются через `git`).

Владелец проекта выполняет

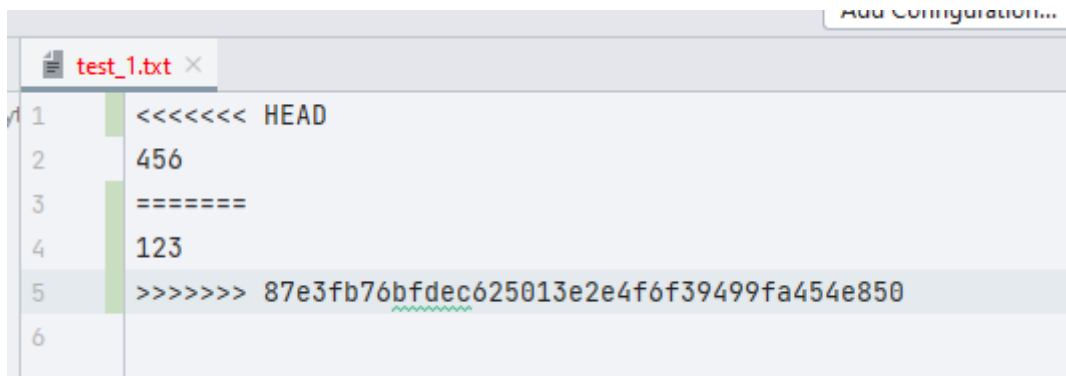
git pull

и убеждается (команда `ls`), что файл `test1.txt` попал в его локальный репозиторий.

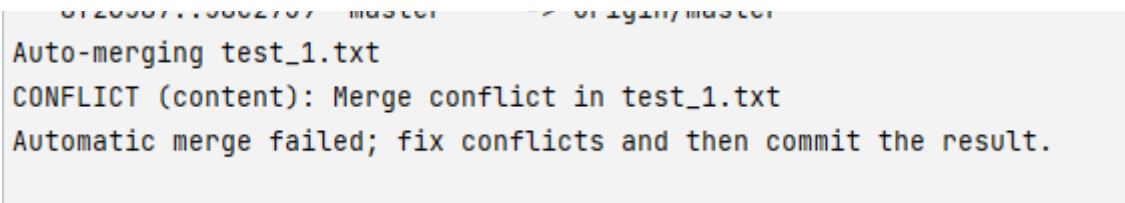
Разрешение конфликтов

Конфликт возникает, когда участники проекта одновременно вносят разную информацию в один файл.

1. Создадим конфликт на стороне владельца проекта:
 - а. Второй участник проекта начинает работу с файлом `test.txt` и записывает в первой строке – 123. Выполняет `add`, `commit`, `push`.
 - б. Владелец проекта в этот же файл (`test.txt`) вносит в первую строку – 456. Выполняет `add`, `commit`, `pull` и получает конфликт

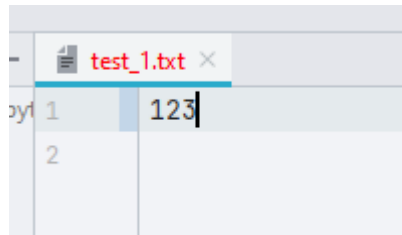


И в консоли:



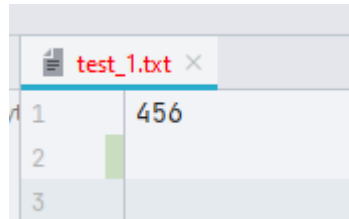
2. Участники проекта теперь должны договориться о том, какая информация является верной:

- а. Если верной оказывается 123, то владелец проекта редактирует файл до состояния



И выполняет add, commit, pull

- а. Если верной оказывается 456, то владелец проекта редактирует файл до состояния



и выполняет add, commit, push

3. Самостоятельно создать и устранить конфликт на стороне второго участника.
4. Поменяется ролями и выполнить все действия по совместной работе сначала.

!!!Внимание!!! По окончанию работы с удаленным репозиторием (в конце урока) необходимо удалить ssh-ключ на github, локальный файл с ключом не удалять.