Методические указания

Настройка Git на рабочей станции (готовность для работы)

# Устанавливаем Git

Если раньше вы не работали с Git, сперва его нужно установить. Способы зависят от операционной системы вашего компьютера.

## Установка в Linux

Используйте обычный менеджер пакетов вашего дистрибутива. Откройте терминал и введите подходящие команды.

* Если у вас 21 или более ранняя версия Fedora, используйте yum install git.
* Для 22 и последующих версий Fedora вводите dnf install git.
* Для дистрибутивов, основанных на Debian, например, Ubuntu, используйте apt-get: sudo apt-get install git.

Полный список команд для различных дистрибутивов можно посмотреть [здесь](https://git-scm.com/download/linux).

## Установка на macOS

1. Скачиваем Git со страницы [проекта](https://git-scm.com/download/mac). ()
2. Запускаем загруженный файл.
3. Система может показать окно с ошибкой, где будет написано, что файл скачан с неавторизованного сайта и инсталлятор не может быть запущен. В таком случае нужно зайти в «Системные настройки» — «Безопасность» (Security and Privacy), в появившемся окне будет сообщение об ошибке и кнопка Open anyway (Всё равно открыть). Нажимаем.
4. Система покажет окно, уточняющее хотите ли вы запустить установку. Подтверждаем действие.
5. Установщик проведёт через все необходимые шаги.

## Установка в Windows

Скачайте exe-файл инсталлятора с [сайта Git](https://git-scm.com/download/win) и запустите его. Это Git для Windows, он называется msysGit. Установщик спросит добавлять ли в меню проводника возможность запуска файлов с помощью Git Bash (консольная версия) и GUI (графическая версия). Подтвердите действие, чтобы далее вести работу через консоль в Git Bash. Остальные пункты можно оставить по умолчанию.

Проверим, что Git установлен

После того, как все действия по установке завершены, убедимся, что Git появился в системе компьютера. Откройте терминал и введите git --version, должна появиться текущая версия программы на вашей машине. Эта проверка подходит для всех операционных систем.

[Пример на MacOS с использованием terminal ]

Команда git, которая выводит справку по git



Проверим версию git с помощью команды git --version

Пример: на моем Mac установлена версия git 2.27.0



# Настройка Git

После того как Git появился на компьютере, нужно ввести свои данные, а именно имя и адрес электронной почты. Ваши действия в Git будут содержать эту информацию.

Откройте терминал и используйте следующую команду, чтобы добавить своё имя: git config --global user.name "ваше имя"

Для добавления почтового адреса вводите: git config --global user.email адрес

Обратите внимание, что в командах, указанных выше, есть опция --global. Это значит, что такие данные будут сохранены для всех ваших действий в Git и вводить их больше не надо. Если вы хотите менять эту информацию для разных проектов, то в директории проекта вводите эти же команды, только без опции --global.



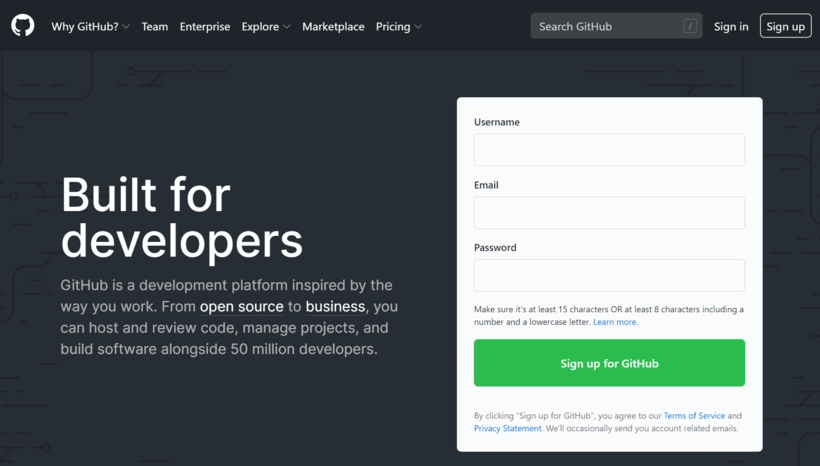
\*Обратите внимание, что у меня уже есть GitHub аккаунт, который называется AstroRacoon и привязан он к почте waleriazl@gmail.com

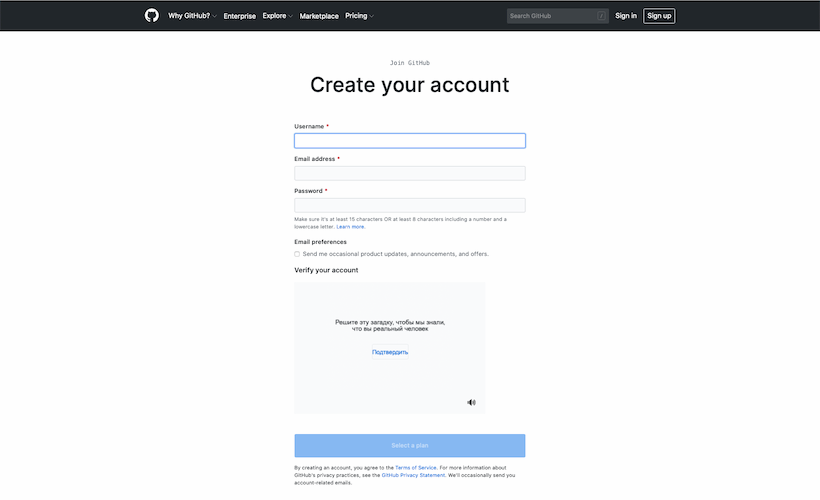
# Регистрация на GitHub

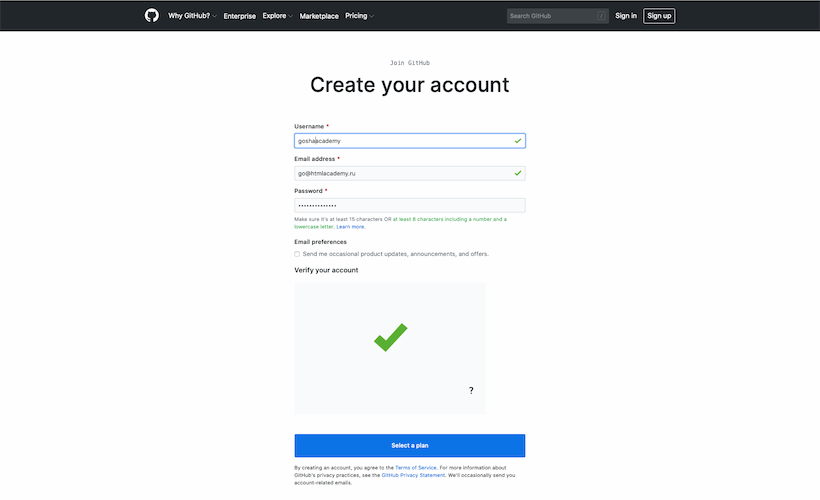
Что такое GitHub?

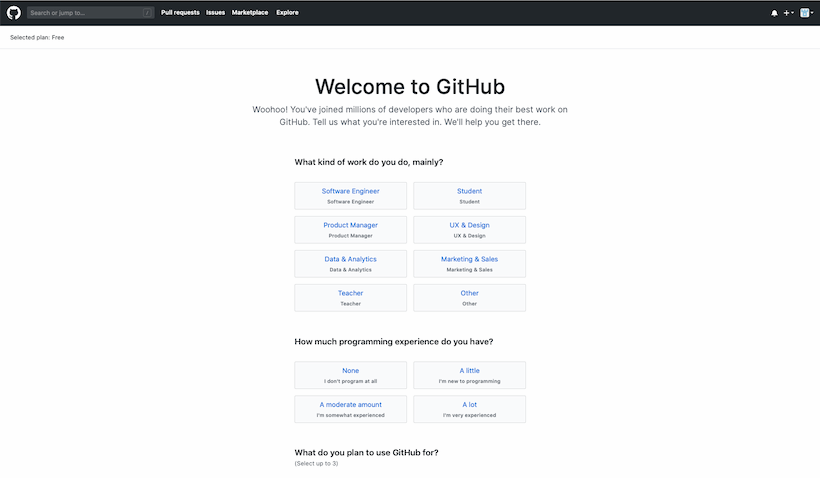
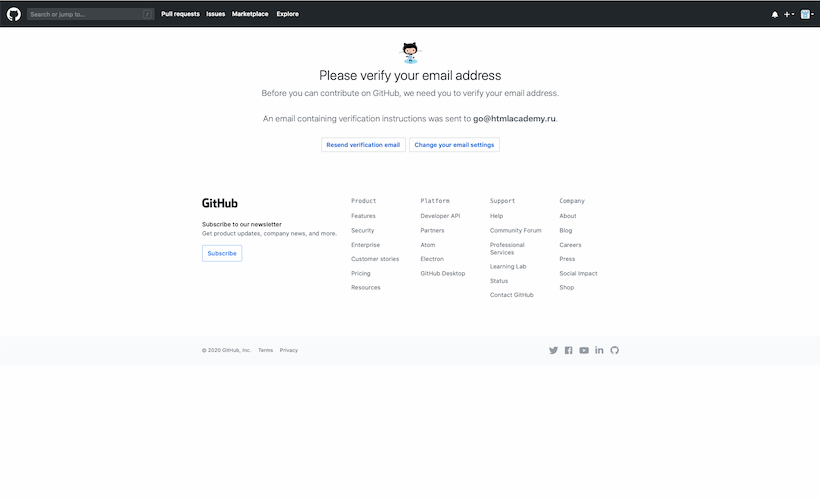
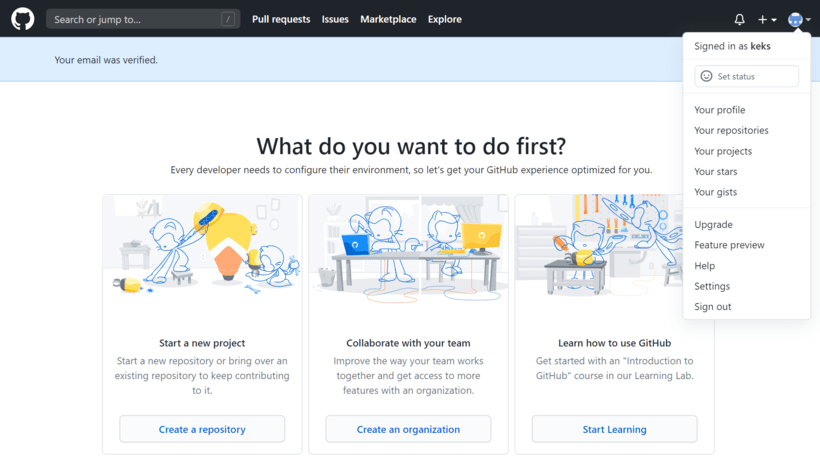
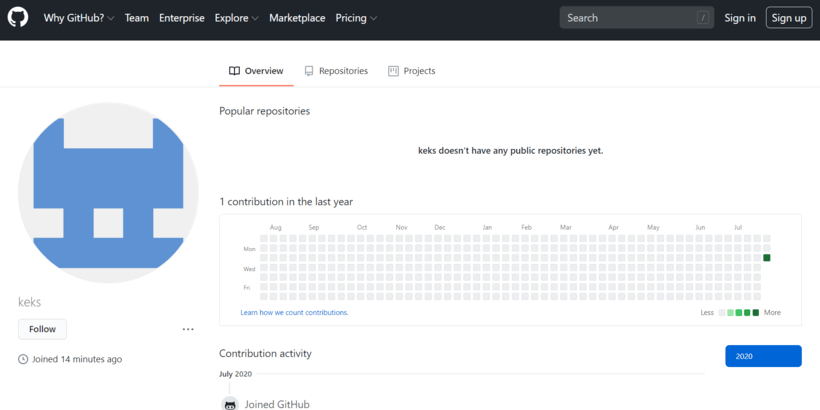
**GitHub** — веб-сервис, который основан на системе Git. Это такая социальная сеть для разработчиков, которая помогает удобно вести коллективную разработку IT-проектов. Здесь можно публиковать и редактировать свой код, комментировать чужие наработки, следить за новостями других пользователей. Именно в GitHub работаем мы, команда Академии, и студенты интенсивов.

Чтобы начать работу с GitHub, нужно зарегистрироваться на сайте, если вы ещё этого не сделали. За дело.

1. Переходим на [сайт GitHub](https://github.com).Cтартовая страница GitHub.
2. Для начала регистрации:

Нажимаем кнопку Sign up (зарегистрироваться), попадаем на страницу регистрации, где вводим обязательные данные: имя пользователя, адрес электронной почты и пароль. После заполнения полей проходим верификацию. Первый шаг регистрации профиля на стартовой странице GitHub.

После заполнения данных и успешного прохождения верификации нажимаем на кнопку Select a plan.Второй шаг регистрации профиля на стартовой странице GitHub.

1. Третий шаг — небольшой опрос от GitHub, который вы можете пройти, заполнив все поля и нажать Submit или пропустить, нажав skip this step.Опрос на третьем шаге регистрации.
2. После прохождения всех этапов на сайте, на указанный при регистрации ящик вам придёт письмо от GitHub. Откройте его и подтвердите свой почтовый адрес, нажав Verify email address (подтвердить электронный адрес) или скопируйте вспомогательную ссылку из письма и вставьте её в адресную строку браузера.Подтверждение электронного адреса.
3. После верификации GitHub предложит создать новый репозиторий, организацию или узнать больше о GitHub. Этот пункт пока можно пропустить и перейти в профиль.Переход в ваш профиль.Так выглядит ваш профиль после регистрации.

Теперь у вас есть профиль на GitHub. Урааа!!!

# Устанавливаем SSH-ключи

Git установлен, профиль на GitHub создан. Осталось добавить SSH-ключ и можно приступать к работе с проектом.

Что такое **SSH-ключ** и зачем он нужен?

Чтобы работать со своего компьютера с GitHub, иметь доступ к проектам, хранящимся на сервисе, выполнять команды в консоли без постоянного подтверждения пароля, нужно пройти авторизацию у сервера. В этом помогают SSH-ключи.

Каждый SSH-ключ содержит пару: открытый (публичный) и закрытый (приватный) ключ. Открытый ключ отправляется на сервер, его можно не прятать от всех и не переживать, что кто-то его увидит и украдёт. Он бесполезен без своей пары — закрытого ключа. А вот закрытый ключ — секретная часть. Доступ к нему должен быть только у вас.

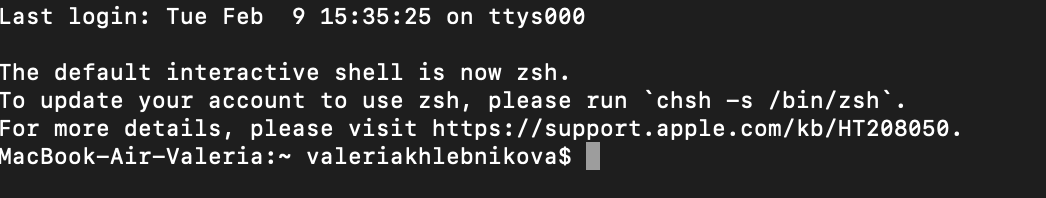
Вы отправляете какую-то информацию на сервер, где хранится ваш публичный ключ, сервер понимает, что вы это вы, то есть идентифицирует именно вас, и даёт вам какой-то ответ. И только вы можете расшифровать этот ответ, потому что только у вас есть подходящий закрытый ключ. Получается что-то вроде связки логин-пароль только намного безопасней. Ваш пароль кто-то может узнать или подобрать, а чтобы получить ваш приватный SSH-ключ, злоумышленнику придётся взломать ваш компьютер.

Чтобы пройти авторизацию по SSH-ключу, его надо сгенерировать или найти уже ранее созданный ключ на своём компьютере.

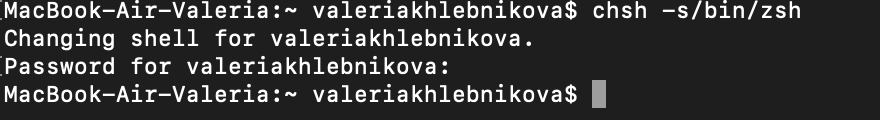
Сначала проверим, есть ли уже на компьютере ключ. По умолчанию SSH-ключи хранятся в каталоге ~/.ssh, поэтому нужно проверить содержимое этого каталога.

1. Открываем консоль.
2. Вводим cd ~/.ssh, чтобы перейти в нужный каталог.

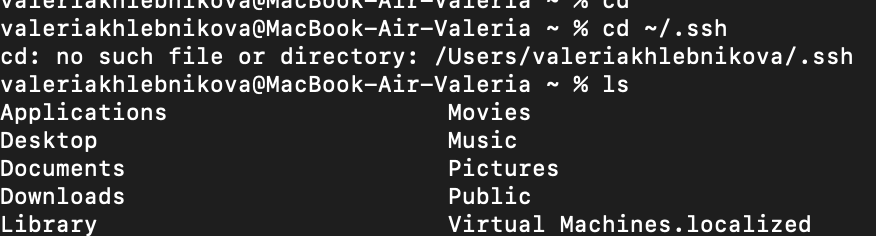
Может потребовать обновить, как в моем случае:



Выполните следующие действия:



Вводим cd ~/.ssh видим, что такой каталог отсутствует



Если таких файлов или даже каталога .ssh у вас нет (как у меня на скриншотах), вы можете их сгенерировать. Для этого делаем следующее.

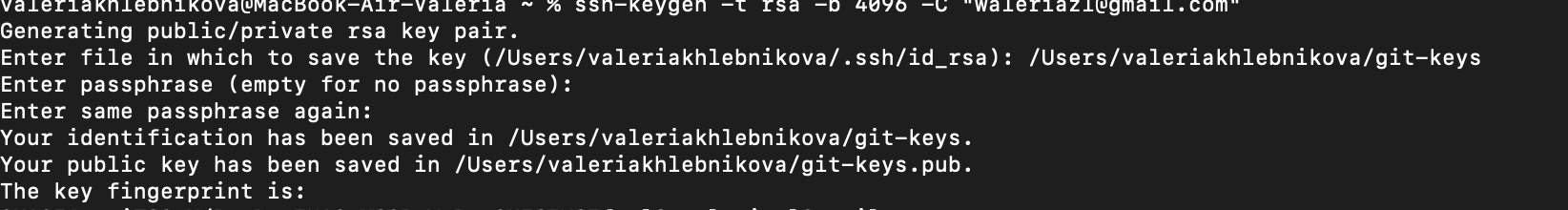
Открываем консоль и вводим команду:

ssh-keygen -t rsa -b 4096 -C "your\_mail@example.com"

Указываем тот адрес электронной почты, который вводили при регистрации на GitHub.

Далее нужно указать расположение файла для сохранения ключа. Если вы не введёте путь до файла и просто нажмёте Enter, ключ сохранится в файле, указанном в скобках. В моем случае указано: /Users/valeriakhlebnikova/git-keys

Теперь нужно установить пароль к вашему ключу и дважды ввести его. Если вы не хотите вводить пароль каждый раз, когда используете ключ, пропустите этот шаг, нажав «Enter», и ничего не вводите.

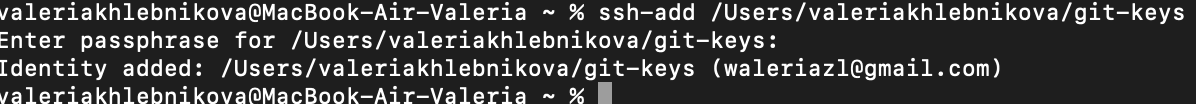


Скриншот обрезан, дальше есть продолжение.

Добавляем ключ в ssh-agent (сгенерированный или уже существующий). Проверяем доступность ключа командой eval "$(ssh-agent -s)" и добавляем с помощью ssh-add ~/.ssh/your\_key\_name, где указываем верный путь до файла с ключом и его имя.



Добавляем с помощью ssh-add



**Несколько важных примечаний:**

* Если вы захотите переименовать ключ, могут возникнуть проблемы. Их можно решить, добавив в ~/.ssh/config связь ключа с доменом.
* Если у вас Windows и вы пользуетесь программой Cmder, возможны проблемы с командой eval "$(ssh-agent -s)". Может появиться такое сообщение об ошибке: «eval не является внутренней или внешней командой, исполняемой программой или пакетным файлом».

В Сmder для запуска ssh-agent можно использовать команду start-ssh-agent.

Если проблема осталась, рекомендуем работать в Git Bash.

* Если у вас macOS Sierra версии 10.12.2 и выше, нужно изменить ваш ~/.ssh/config файл, чтобы автоматически загрузить ключи в ssh-agent и хранить пароли.
* Host \*
* AddKeysToAgent yes
* UseKeychain yes

IdentityFile ~/.ssh/id\_rsa

Вы можете добавить свой приватный ключ в ssh-agent и сохранить пароль к нему с помощью команды ssh-add -K ~/.ssh/id\_rsa. Если у вашего ключа другое имя, не забудьте заменить id\_rsa в команде на правильное название.

Если у вас Linux, может понадобится переназначить для ~/.ssh права доступа командой chmod 700 ~/.ssh/

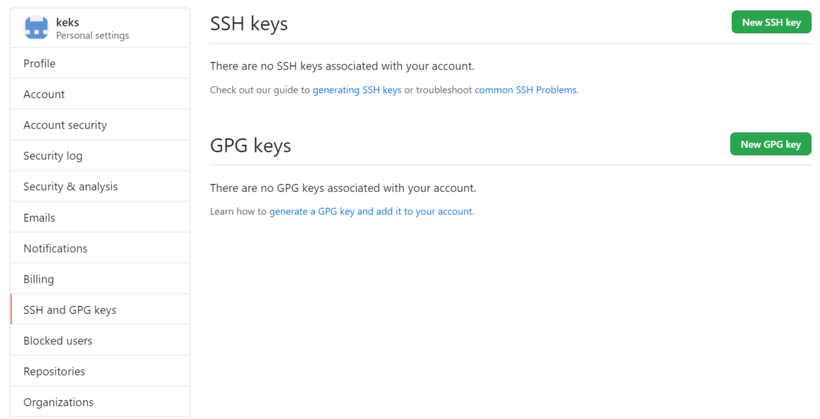
После того как создан ключ, его нужно добавить на GitHub. Для этого копируем его содержимое с помощью одной из следующих команд:

Если вы на Windows clip .

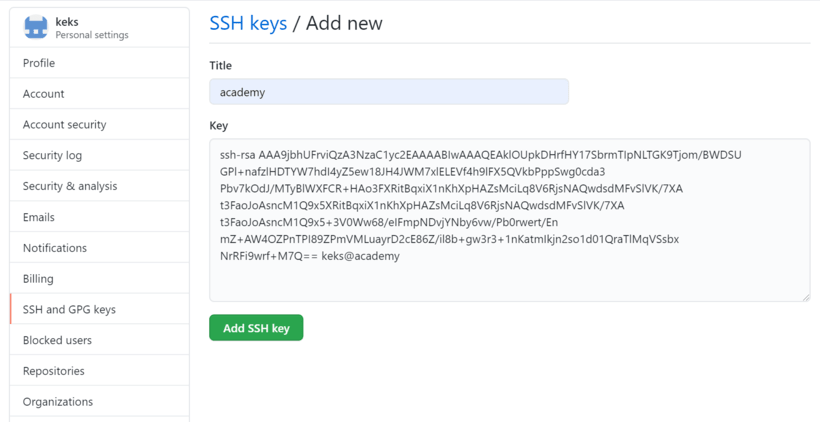
Для пользователей macOS pbcopy .

На Linux используйте sudo apt-get install xclip, чтобы установить необходимый для копирования пакет xclip, а затем введите xclip -sel clip . Или введите команду cat ~/.ssh/id\_rsa.pub, контент документа появится прямо в консоли и вы сможете скопировать ключ оттуда.

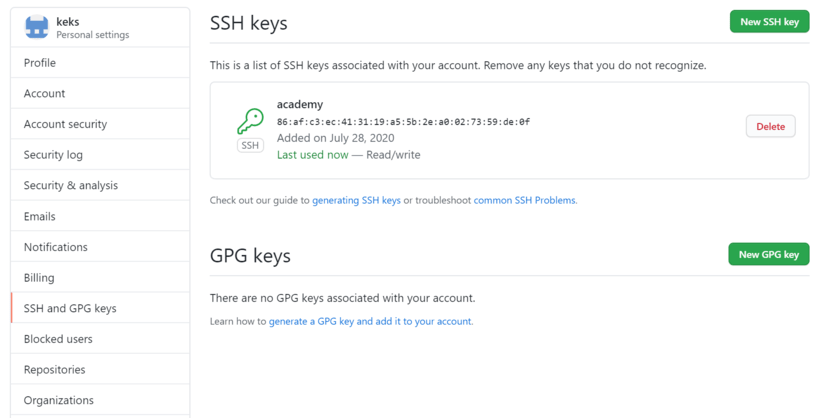
Можно пойти другим путём, открыть файл id\_rsa.pub прямо в папке и просто скопировать содержимое оттуда.

Переходим на [страницу для работы с ключами](https://github.com/settings/keys) в вашем профиле на GitHub.Страница с настройками ключей в вашем профиле.

Нажимаем кнопку New SSH key (новый SSH-ключ). Вводим имя ключа (можно придумать абсолютно любое) в поле Title (название), а в Key (ключ) вставляем сам ключ из буфера обмена. Теперь нажимаем Add SSH key (добавить SSH-ключ).

Добавляем в свой профиль SSH-ключ.

Если всё сделано верно, в списке появится новый ключ.

Успешно добавленный ключ.

Теперь, наконец-то, мы можем начать работу с самим проектом.