

КАК ВЫЛОЖИТЬ СВОЙ ПРОЕКТ НА GITHUB?

🏠 (<https://proweb63.ru/>) / [Блог по веб разработке \(help/\)](#) / [Git \(help/git/\)](#) / [Как выложить свой проект на GitHub? \(help/git/create-github-repo\)](#)

Как выложить свой проект на GitHub?

Если вы, как и я, решили освоить git, то скорее всего у вас возникнет желание разместить свой проект на GitHub.com для публичного доступа.

Как новичку мне было сложно понять, каким образом создать связь между репозиториями. То есть локально у нас есть набор файлов проекта.

Нужно каким-то образом выложить их на гитхаб. По сути это обычное копирование, но в git это работает не так очевидно. Пробившись несколько дней, я написал свою пошаговую инструкцию для новичков.

Два способа

Существует два способа для связи с гитхабом. Они далеко не так очевидны для новичков. Основной нюанс заключается в том, какой именно будет использован гитхабовский репозиторий: совершенно пустой (не инициализированный) или уже рабочий. Именно это будет определять два разных варианта работы.

Первый способ подходит для уже инициализированных удаленных репозиториях. Для связи GitHub с локальной копией, вначале **следует клонировать гитхабовский репозиторий как локальный**. Только после этого можно изменить локальный и отправить его на удалённый.

Второй способ используется для совершенно пустого гитхабовского репозитория. Если уже есть локальный репозиторий, то **можно вручную связать его с гитхабом без клонирования**.

Регистрация на GitHub.com

Вначале нужно зарегистрироваться на GitHub.com (<https://github.com/>). Процедура простая, не будем на ней останавливаться. После этого нужно создать репозиторий. В поле «Repository name» следует указать имя, которое в будущем будет совпадать с каталогом проекта. Это удобно, хотя, локальный каталог может быть любым. Пусть проект называется «demo».

Owner * Repository name *

MaxiDeBox / mydemo ✓

Great repository names are short and memorable. Need inspiration? How about **automatic-spoon**?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

Здесь очень важный момент. Если вы отметите любые опции (добавить .gitignore, лицензию и/или readme), то фактически это будет означать инициализацию репозитория. То есть это уже не пустой репозиторий, а наполненный и инициализированный. В этом случае для связки придётся вначале клонировать удаленный. Это первый способ.

Если же вы не отметили ни одной опции, то гитхаб создаст совершенно пустой репозиторий без инициализации. И вот такой репозиторий можно связать с уже существующим локальным. Это второй способ.

После создания репозиторий получит уникальную ссылку, например: `https://github.com/USER/demo.git` - где USER — ваш логин в гитхабе. Эта ссылка потребуется для работы.

Установка Git

Я всё ставил с официального сайта Git (<http://git-scm.com/>). Там же на сайте есть руководство Book (<http://git-scm.com/book/ru/v1>), где описаны все начальные шаги. Поэтому будем считать, что git установлен и настроен.

Рассмотрим два способа отдельно.

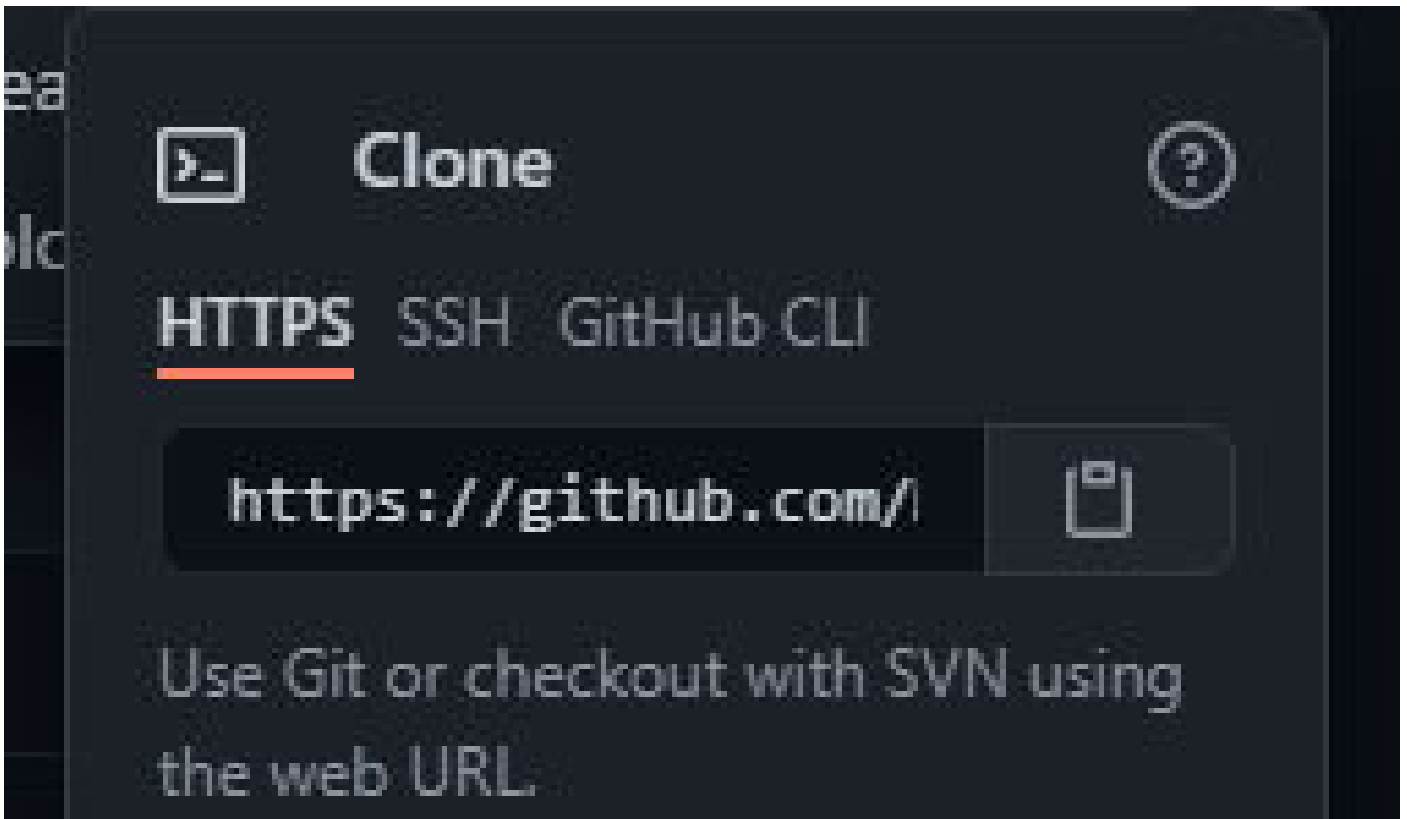
Первый способ

Наш проект размещается в каталоге `/demo/`. Было бы неплохо сохранить этот каталог и для дальнейшей работы. Для того, чтобы обезопасить себя, переименуем его `/demo-temp/` любым файловым менеджером. То есть теперь каталога `/demo/` у нас нет.

Запускаем **git-bash** и клонируем удаленный репозиторий на локальную машину:

```
git clone https://github.com/USER/demo.git
```

— где **USER** — ваше имя на гитхабе. Ссылку также можно скопировать со страницы репозитория в поле «HTTPS clone URL».



Появится каталог `/demo/` который создал `git`. Он должен быть пустой, кроме подкаталога `«.git»` — это служебный каталог, и его трогать не нужно.

Теперь копируем в каталог `/demo/` содержимое нашего проекта, которое мы сохранили в `/demo-temp/`. Всё, что мы сюда скопируем, будет выложено на GitHub, поэтому желательно удалить все ненужные файлы. Если какие-то служебные файлы требуются для проекта, то их можно указать в файле `.gitignore`.

Выполняем команду

```
git add .
```

Это добавит все файлы для отслеживания `git`'ом. Проверить состояние (до и после `add`) можно командой

```
git status
```

Теперь делаем коммит (сообщение любое):

```
git commit -m "Add project"
```

Пока мы работаем с локальным репозиторием. То есть нужно настроить его так, чтобы он оказался в том виде, который отправится на гитхаб.

Для удобства можно сразу выставить метку версии (опять же любой вариант):

```
git tag v1.0
```

Теперь можно отправить изменения на гитхаб.

Теперь нужно обновить метки на удаленном репозитории (если вы их задали).

Для настройки **push**, если требуется, пишем:

Для связи с GitHub'ом следует указать удаленный репозиторий:

4/8