Как выложить свой проект на GitHub.com

16 января 2015 г.[Git. GitHub](https://maxsite.org/category/git)

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.

Способ для IntelliJ IDEA

https://www.youtube.com/watch?v=kO5u0PFdHUQ

Два способа GitBash

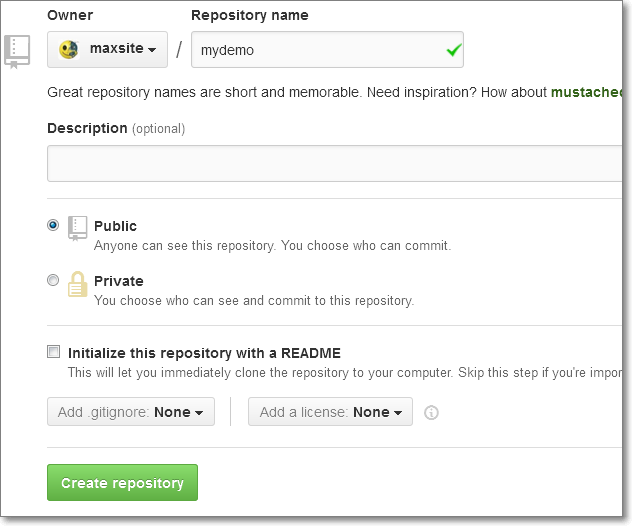
Существует два способа для связи с гитхабом. Они далеко не так очевидны для новичков. Основной нюанс заключается в том, какой именно будет использован гитхабовский репозиторий: совершенно пустой (не инициализированный) или уже рабочий. Именно это будет определять два разных варианта работы.

Первый способ подходит для уже инициализированных удаленных репозиториев. 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».



Здесь очень важнный 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 инициализировнный. 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 первый 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 начальне шаги. Поэтому будем считать, что git установлен и настроен.

Рассмотрим два способа отдельно.

**Первый способ**

1. Создать новый репозиторий на git-е.

2. Все шаги (которые написаны ниже) по копированию проекта на git будут написаны на сайте github на первой странице моего нового репозитория. Единственное отличие, что в последней команде (в конце) нужно будет прописать ветвь master *(как тут написано)*, а не предложенную мне в последнем новом проекте ветвь main:

2.1. открываем gitbush у меня на компе на папке выкладываемого проекта (если я на своем компе сделал копию уже существующего на github проекта то нужно из нее удалить папку “.git”)

2.2. git init //создаем у меня на компьютере в директории, в которой сейчас находится gitbush, папку “**.git**”

2.3. git add . //загоняем все файлы из директории, в которой сейчас находится gitbush, в промежуточную фазу Index

2.4. git commit -m "Init" //из промежуточной фазы Index переносим все находящиеся файлы в конечную фазу HEAD + комментарий который мы написали в кавычках

Теперь для проекта git работает и его можно использовать по своему усмотрению: добавлять версии, смотреть логи, делать ветки и т.п.

Для связи с GitHub'ом следует указать удаленный репозиторий:

2.5. git remote add origin https://github.com/USER/demo.git //соединяемся с необходимым нам удаленным репозиторием, находящимся на сервере git-a (адрес написан вверху первой страницы моего нового репозитория)

2.6. git push -u origin master //копируем информацию из HEAD в удаленный репозиторий на git-e

Этот код указывает адрес удаленного репозитория и отправляет все изменения на гитхабовский сервер. Если мы зайдем на страницу репозитория на гитхабе, то также увидим свой проект.

Второй 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 вылождено 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.

git push

Git может выдать сообщение, что команда **push** не настроена по-умолчанию. Пока не обращайте внимания на это сообщение. Нужно будет внести изменения в конфигурацию позже.

Git потребует ввести логин и пароль для GitHub.com. Учитывайте, что пароль полностью скрывается, поэтому не будет видно даже «звездочек». После ввода нажимаем Enter и git выполнит обновление удаленного репозитория.

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

git push --tags

Здесь также нужно будет ввести логин и пароль.

Всё, синхронизация выполнена! Временный каталог /demo-temp/можно удалить или сохранить как старый резервный вариант.

Если мы зайдем на страницу репозитория на гитхабе, то увидим свой проект.

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

git config --global push.default simple