Краткая инструкция как заливать на gitHub

Содержание

Быстрый старт (для нетерпеливых)

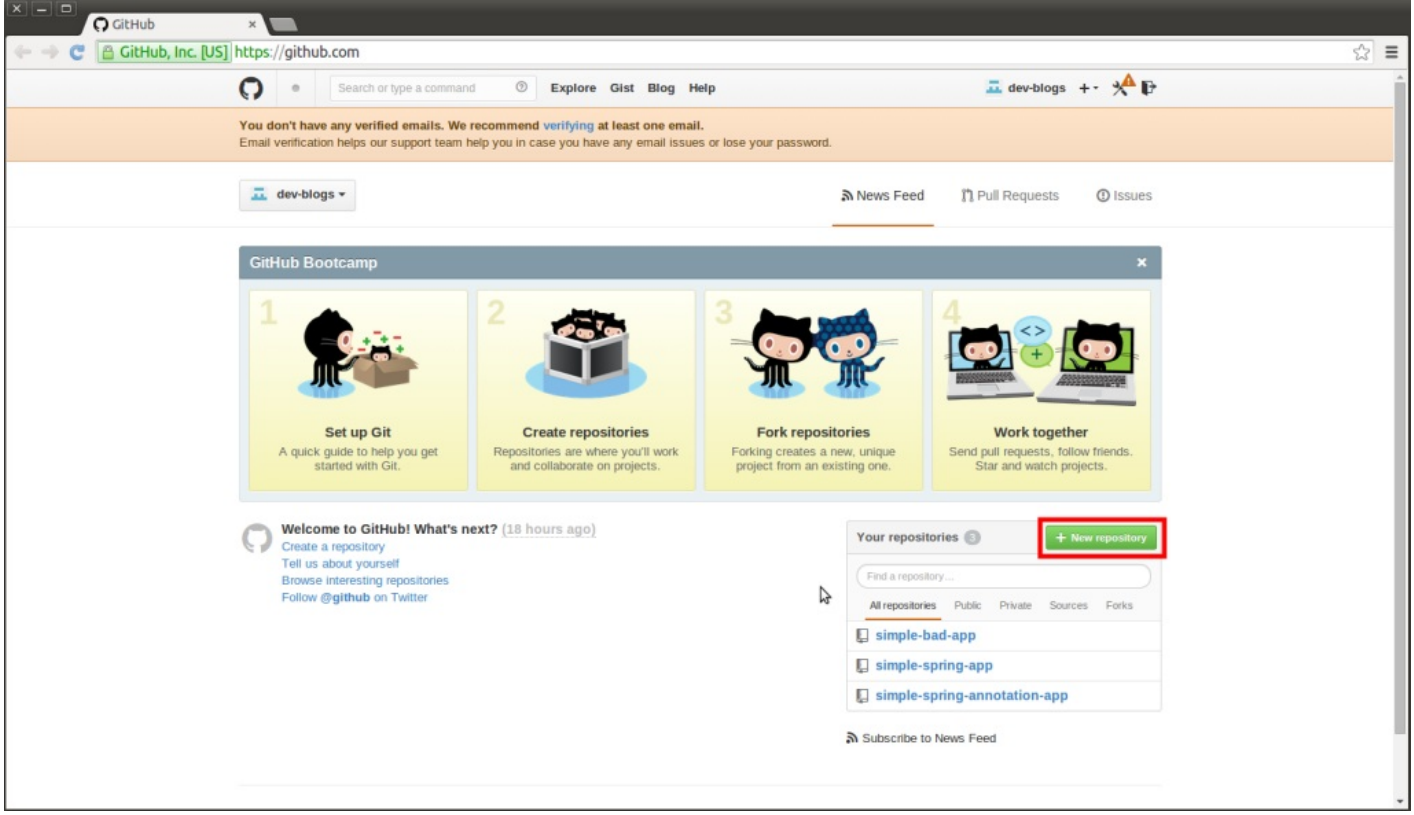
Если нет времени читать весь пост, а надо сию секунду залить свой код на удаленный репозиторий не вникая во все тонкости, то делается это так:

```
git init
git add your_file
git commit -m "first commit"
git remote add origin https://github.com/you_repository/you_project
git push -u origin master
```

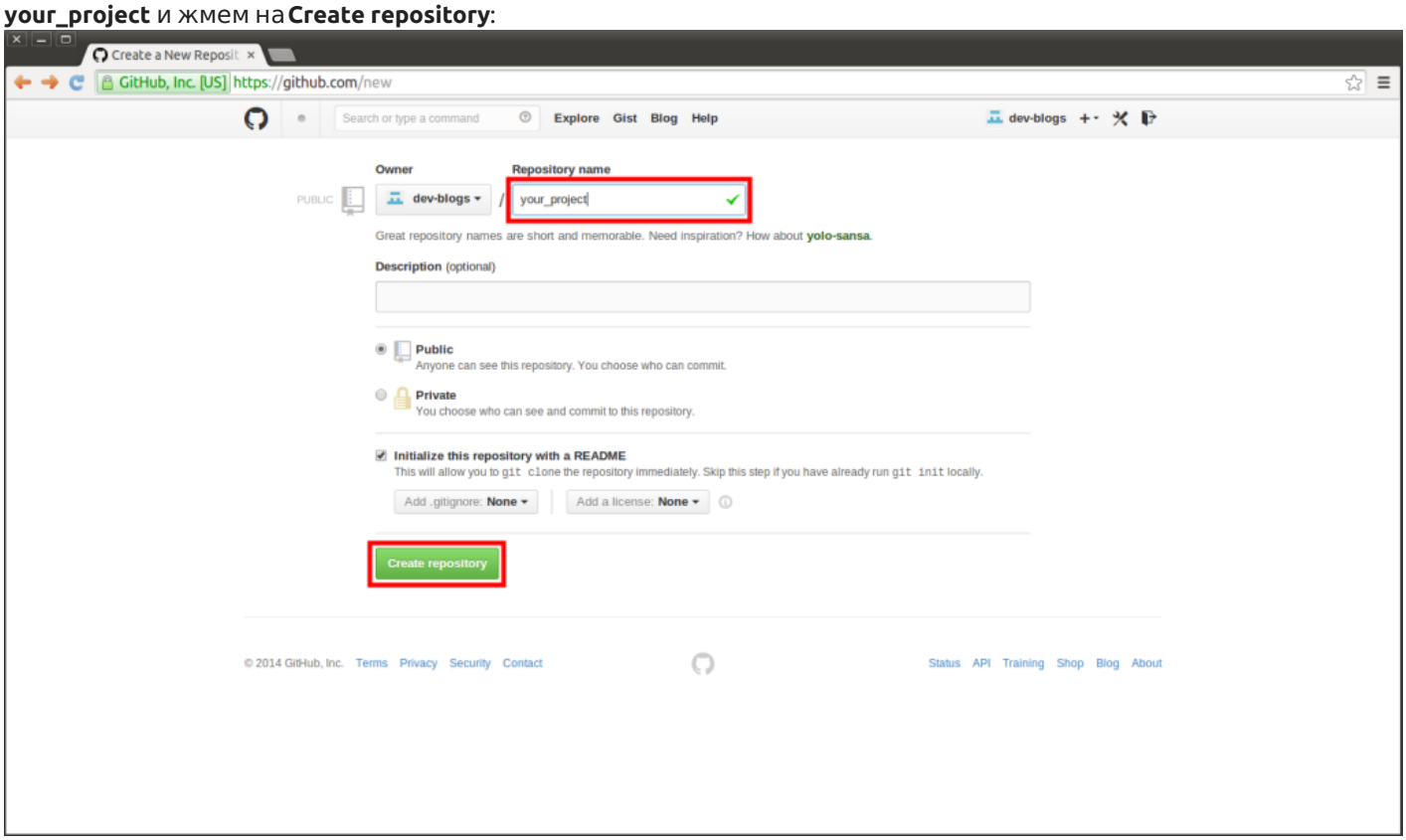
Все!

Создание репозитория на гитхабе

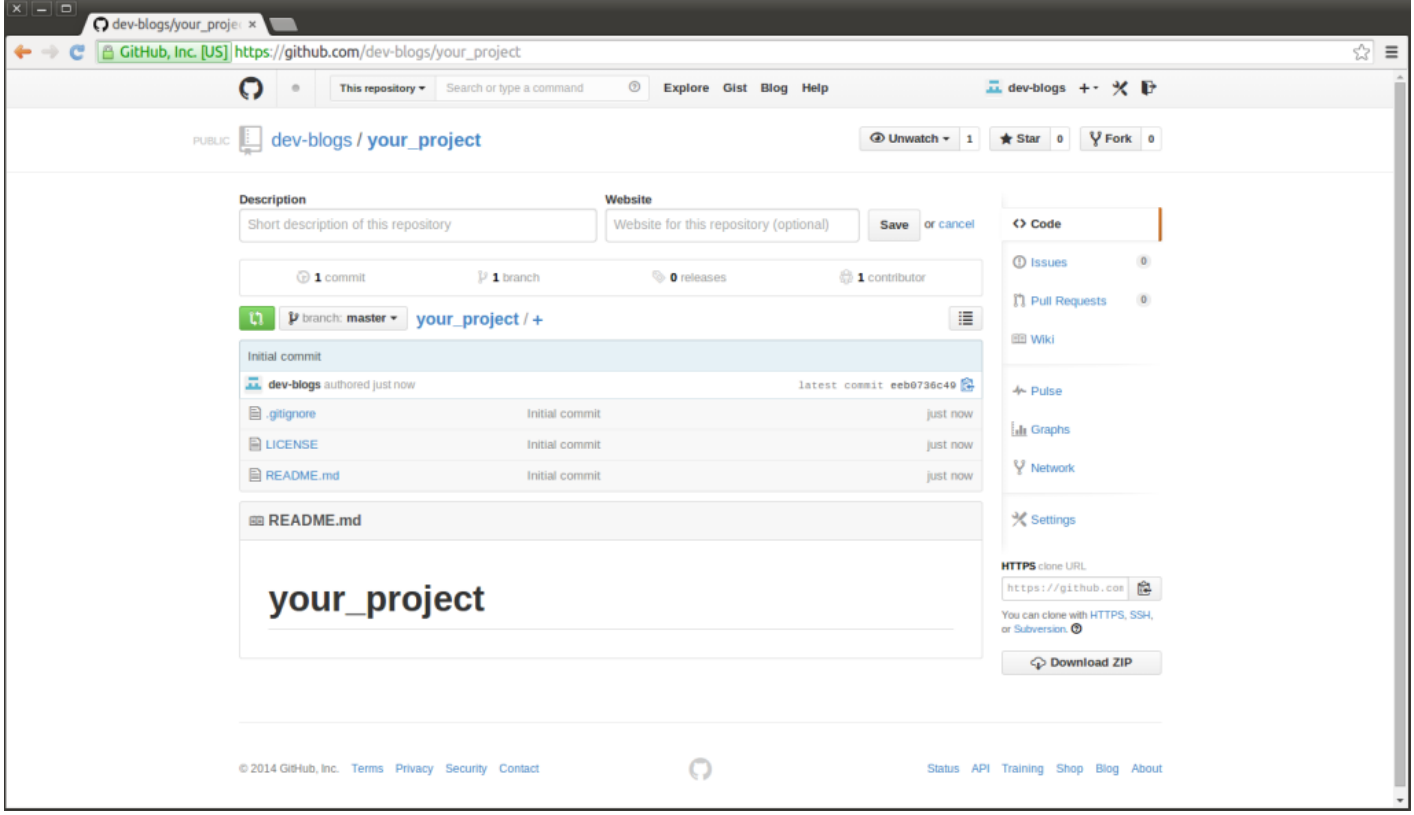
Теперь поговорим более подробно о гите и как заливать локальный репозиторий на удаленный. Начнем с создания удаленного репозитория на популярном сервисе https://github.com. Чтобы залить проект на**gitHub** нужно сначала создать в gitHub аккаунт и залогиниться.
После этого жмем на **+ New repository**:



Появится страница **Create a New Repository**. В поле **Repository name** вводим имя репозитория, например **your_project** и жмем на **Create repository**:



Появится созданный репозиторий **your_project**:



Если это будет java-приложение, то нужно за комментировать ***.jar** в файле **.gitignore**. Открываем на редактирование файл **.gitignore** и за комментируем, в нашем случае, седьмую строчку:

`.gitignore`

```
1  *.class
2
3  # Mobile Tools for Java (J2ME)
4  .mtj.tmp/
5
6  # Package Files #
7  #*.jar
8  *.war
9  *.ear
10 # virtual machine crash logs, see http://www.java.com/en/download/help/error_hotspot.xml
11 hs_err_pid*
```

Подготовка локального `git` репозитория

Подготовить локальный репозиторий можно двумя способами: создать репозиторий с нуля с последующим переносом изменений в удаленный репозиторий и сделать клон удаленного репозитория.
Первый способ создание локального репозитория с нуля командой**git init**.
Создадим проект на локальной машине с таким именем:

```
mkdir your_project
```

перейдем в этот каталог:

```
cd your_project
```

Выполним команду **git init** которая инициирует локальный репозиторий:

```
git init
```

Дальше можно добавлять файлы в локальный репозиторий.
Второй способ. Сделать на локальной машине клон удалённого репозитория командой**git clone**:

```
git clone https://github.com/you_account/your_project
```

После этой команды у нас появится новый каталог в котором находится копия удаленного репозитория, а все файлы которые в нем находятся будут отслеживаться гитом. Тут очень важный момент именно **копия всего репозитория**, а не снимок текущего состояния удаленного репозитория. В отличие от обычного снимка удаленного репозитория, например как в **SVN** мы, будучи скопировав удаленный репозиторий, можем покопаться в его истории, посмотреть все его правки, кто и когда вносил изменения, какие у него ветки, то есть у нас на машине полноценный репозиторий который теперь не зависит от удаленного репозитория с

которого был склонирован.

Подготовка локального файла

После того как появился локальный репозиторий, добавим в него джава класс. Перейдём в каталог, который отслеживается репозиторием и создадим какой-нибудь файл::

`TestGitHub.java`

```
1  public class TestGitHub {
2      public static void main(String [] args) {
3          System.out.println("Test hibHub");
4      }
5  }
```

Помещение файла в репозиторий

После того как мы создали файл его надо подготовить для фиксации и зафиксировать в репозитории, то есть закомитить. **Подготовить для фиксации** это означает, что его надо **проиндексировать** командой **git add**:

```
git add *
```

Проиндексированный файл это еще не означает, что он закомичен, это означает, что он готов для коммита в репозиторий, а сам коммит выполняется командой **git commit**:

```
git commit -m "create project"

[master 412c945] create project
 Committer: zheka <zheka@zheka-Vostro-3560.(none)>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 5 insertions(+)
 create mode 100644 src/TestGitHub.java
```

Если гит ругнется как показано ниже:

```
*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'computer-name.(none)')
```

значит перед тем как коммитить изменения надо сообщить гиту свое имя и почту. Это нужно сделать так как гит включает эту инфу в каждую фиксированную версию:

```
git config --global user.name "user"
git config --global user.email user@example.com
```

Перенос изменений на удаленный репозиторий

Локальный репозиторий готов, теперь осталось перенести его на удаленный. Переносится репозиторий командой **git push**, но прежде чем переносить мы должны выяснить со сколькими репозиториями мы работаем и выбрать из списка тот, в который мы хотим перенести наши изменения. Для того, чтобы увидеть все удаленные репозитории нужно выполнить команду **git remote -v**:

```
git remote -v
origin  https://github.com/your_account/your_project (fetch)
origin  https://github.com/your_account/your_project (push)
```

Но мы увидим удаленные репозитории только в том случае, если мы с клонировали его командой **git clone**, в случае если мы создали локальный репозиторий командой **git init**, то мы ничего не увидим, в этом случае нам надо добавить удаленный репозиторий, это будет далее. А сейчас, допустим у нас есть клон удалённого репозитория. Выполнив команду **git remote -v** мы увидим url адреса и короткое имя для удалённых репозиториев с которыми мы работаем. В данном случае мы работаем с одним удаленным репозиторием, которому присвоено короткое имя по умолчанию **origin**, который находится по адресу **https://github.com/your_account/your_project** как для фетча, так и для пуша.
Теперь можем переносить все изменения для репозитория **origin** командой **git push**:

```
git push origin master

Username for 'https://github.com': dev-blogs
Password for 'https://dev-blogs@github.com':
To https://github.com/dev-blogs/your_project
   eeb0736..412c945  master -> master
```

После этого github запросит имя юзера и пароль.
То что мы сейчас сделали мы запушили (выложили) наши локальные изменения на удаленный репозиторий у которого айдишник **origin** в ветку **master**.

Добавление удаленных репозиториев

Если мы создали репозиторий командой **git init**, то чтобы перенести изменения на удаленный репозиторий, нам надо его добавить командой **git remote add** и придумать ему уникальное имя. Вот как добавляется удаленный репозиторий:

```
git remote add ourRep https://additional_git_address/your_account/your_project
```

Мы задали удаленный репозиторий с коротким именем **ourRep**, который располагается по адресу **https://additional_git_address/your_account/your_project**. Если выполним команду **git remote**, то увидим url адреса для пуша и фетча только что добавленного удаленного репозитория:

```
git remote -v
ourRep  https://additional_git_address/your_account/your_project (fetch)
ourRep  https://additional_git_address/your_account/your_project (push)
```

Теперь зальём на него изменения командой **git push**:

```
git push ourRep master

Username for 'https://github.com': dev-blogs
Password for 'https://dev-blogs@github.com':
To https://github.com/dev-blogs/your_project
   eeb0736..412c945  master -> master
```

Ссылки

Git в картинках

Поделиться в социальных сетях

UNIX VCS | dev-blogs | Июнь 30, 2014

← Сборка и настройка tomcat

Доступ к SVN через протокол WebDAV (HTTP) →