

Отчёт по лабораторной №1. Работа с GIT

Подготовка, установка и применение git

Шаян Фаисал НФИбд-02-19

Содержание

1	Цель лабораторной работы №1	4
2	Ход работы лабораторной №1	5
3	Вывод:	10
	Список литературы	11

List of Figures

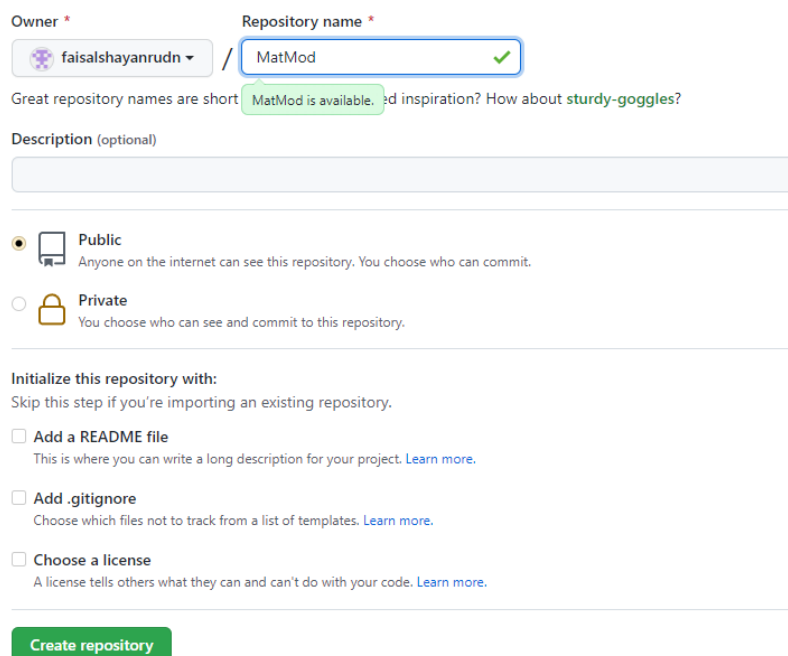
2.1	Создание учетной записи и репозитория	5
2.2	Инициализация репозитория и создание файла	6
2.3	Создание ключа	6
2.4	Добавляем ключ на сайт	7
2.5	Загрузка файлов с дальнейшей отправкой в сетевой репозиторий	7
2.6	Инициализация git-flow и начало релиза, пробуем простые коман- ды для работы с ветками	8
2.7	Завершаем релиз и отправляем изменения в сетевой репозиторий	8
2.8	Объединяем ветки в сетевом репозитории	9

1 Цель лабораторной работы №1

Целью данной лабораторной работы является изучение пространства git, применение различных команд и отработка элементарных действий с системой.

2 Ход работы лабораторной №1

Создаем учетную запись на github.com и репозиторий, в котором в дальнейшем будем работать



The screenshot shows the GitHub repository creation interface. At the top, there are two input fields: 'Owner' with a dropdown menu showing 'faisalshayanrudn' and 'Repository name' with a text input containing 'MatMod'. A green checkmark is visible next to the repository name. Below these fields, a message states: 'Great repository names are short' followed by a green box containing 'MatMod is available.' and the text 'Get inspiration? How about sturdy-goggles?'. Underneath is a 'Description (optional)' text area. The next section is for repository visibility, with 'Public' selected by default (indicated by a radio button and a lock icon) and 'Private' as an option. Below this is the 'Initialize this repository with:' section, which includes three checkboxes: 'Add a README file', 'Add .gitignore', and 'Choose a license'. Each checkbox has a brief description and a 'Learn more' link. At the bottom of the form is a green button labeled 'Create repository'.

Figure 2.1: Создание учетной записи и репозитория

Инициализируем созданный нами локальный репозиторий, после создаем в нем файл `README.md`, с которым будем проводить все дальнейшие действия

```

PS C:\Labs>
PS C:\Labs> git init
Initialized empty Git repository in C:/Labs/.git/
PS C:\Labs> echo "# лабораторные работы" >> README.md
PS C:\Labs> git add README.md
PS C:\Labs>

```

Figure 2.2: Инициализация репозитория и создание файла

Создаем SSH-ключ, который пропишем в настройках на github.com

```

PS C:\Labs> git init
Initialized empty Git repository in C:/Labs/.git/
PS C:\Labs> echo "# лабораторные работы" >> README.md
PS C:\Labs> git add README.md
PS C:\Labs>
PS C:\Labs> git config --global user.name faisalshayanrudn
PS C:\Labs> git config --global user.email "1032189252@pfur.ru"
PS C:\Labs> git commit -m "first commit"
[master (root-commit) 0596912] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
PS C:\Labs> ssh-keygen -C "faisalshayanrudn 1032189063@pfur.ru"
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\User/.ssh/id_rsa):
Created directory 'C:\Users\User/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\User/.ssh/id_rsa.
Your public key has been saved in C:\Users\User/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:95erIOU/iHCbiYqsgcvoqdhqrCvo7fKKJWU43F3uuI0 faisalshayanrudn 1032
189063@pfur.ru
The key's randomart image is:
+----[RSA 2048]-----+
|
|... . o
|o.o. . .S o
|. + + .+ .
|* . . =.o.. o
|B@o. = ..o..
|^\XO+.E . oo.
+----[SHA256]-----+
PS C:\Labs> cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACjBoI+pzuvsvyANKRwn8syXI87fVle5bmQ
DxPFdYIcJ5/MuFcNhJ9QcydUB5zX7jb5l2JscqaXjsq26zCztvAZcrVVY5D7ogryRN1zC0iM
gnT6r3ghkBVszCrm4Y2KjpFbVILmW4Xl1srycqovCWxy4RMXxmVrahujdvV7xC3pYftPQsck
NjNa1G+QvPvsEs5Mb/CLMFT/5g0lrIr7Qat9d/avH6hzcMXMwC0SuxCun7Y8y8h8CTl1PlsS
0p0w2kN6xoKWGnNQBvPpwP6i4BzTgFpi53LuLR4/+mwrIB56XUNdZ3JcdYPSkaOUa2VT1FmI
TnGxtQwsKKQ3LsbQZppD faisalshayanrudn 1032189063@pfur.ru
PS C:\Labs>

```

Figure 2.3: Создание ключа

SSH keys / Add new

Title

key-code

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCBol+pzuvsyaNKRWn8syXlf87fVle5bmQDxPFdYlcJ5/MuFcNhJ9QcydUB5zX7jb
5I2JscqaXjsq26zCztvAZcrVVY5D7ogryRN1zCOiMgnT6r3ghkBVszCrm4Y2KjpFbVilMw4Xl1srycqovCWxy4RMXxmvRahujdV
V7xC3pYftPQScKNUjNaIG+QvPvsEs5Mb/CLMFT/5gOlrIr7Qat9d/aVH6hzcMXMwC0SuXCun7Y8y8h8CTI1PlsS0p0w2kH6XoK
WGnNQ8vPpwP6i4BzTgFpi53LuLR4/+mwrlB56XUNdZ3JcdYPSkaOUa2VT1FmlTnGxtQWsKKQ3LsbQZppD
faisalshayanrudn 1032189063@pfur.ru
```

Add SSH key

Figure 2.4: Добавляем ключ на сайт

Загружаем файлы лицензионного соглашения и gitignore и отправляем эти файлы в сетевой репозиторий.

```
PS C:\Labs> git remote add origin git@github.com:faisalshayanrudn/MatMod.git
PS C:\Labs> wget https://creativecommons.org/licenses/by/4.0/legalcode.txt -O LICENSE
PS C:\Labs> wget https://www.toptal.com/developers/gitignore/api/python -O .gitignore
PS C:\Labs> git add .
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in LICENSE.
The file will have its original line endings in your working directory
PS C:\Labs> git commit -am "add license"
[master 8f3f012] add license
2 files changed, 555 insertions(+)
create mode 100644 .gitignore
create mode 100644 LICENSE
PS C:\Labs> git push -u origin master
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCoQU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 7.72 KiB | 1.93 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:faisalshayanrudn/MatMod.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
PS C:\Labs> git push
Everything up-to-date
PS C:\Labs>
```

Figure 2.5: Загрузка файлов с дальнейшей отправкой в сетевой репозиторий

Используем системы управления версиями и тегами. Создаем ветку, начинаем и завершаем в ней релиз.

```

PS C:\Labs> git flow init
which branch should be used for bringing forth production releases?
- master
Branch name for production releases: [master]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? [] v
Hooks and filters directory? [C:/Labs/.git/hooks]
PS C:\Labs> git branch
* develop
  master
PS C:\Labs> git flow release start 1.0.0
Switched to a new branch 'release/1.0.0'

Summary of actions:
- A new branch 'release/1.0.0' was created, based on 'develop'
- You are now on branch 'release/1.0.0'

Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

    git flow release finish '1.0.0'

PS C:\Labs> echo "1.0.0" >> version
PS C:\Labs> git add .
PS C:\Labs> git commit -am "main: add version"
[release/1.0.0 60108bc] main: add version
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 version
PS C:\Labs> git flow release finish -m "ver 1" 1.0.0
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
Merge made by the 'ort' strategy.

```

Figure 2.6: Инициализация git-flow и начало релиза, пробуем простые команды для работы с ветками

```

PS C:\Labs> git push --all
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 495 bytes | 165.00 KiB/s, done.
Total 5 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To github.com:faisalshayanrudn/MatMod.git
 * [new branch]      master -> master
 * [new branch]      develop -> develop
PS C:\Labs> git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 165 bytes | 82.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:faisalshayanrudn/MatMod.git
 * [new tag]         v1.0.0 -> v1.0.0
PS C:\Labs>

```

Figure 2.7: Завершаем релиз и отправляем изменения в сетевой репозиторий

Выполняем объединение веток с помощью простых команд.

Merge tag 'v1.0.0' into develop #1

Merged faisalshayanrudn merged 1 commit into `master` from `develop` now

Conversation 0 Commits 1 Checks 0 Files changed 0

faisalshayanrudn commented now Owner ...

ver 1 v1.0.0

Merge tag 'v1.0.0' into develop ... a26459c

faisalshayanrudn merged commit 36fb2df into `master` now Revert

Pull request successfully merged and closed Delete branch

You're all set—the `develop` branch can be safely deleted.

Figure 2.8: Объединяем ветки в сетевом репозитории

3 Вывод:

Мы приобрели практические навыки работы с системой контроля версий git и создали свой репозиторий. Изучили пространства git, применили различные команды и отработали элементарные действия с системой.

Список литературы

1. Git Шпаргалка
2. Основы Git
3. Руководство по оформлению Markdown файлов