

Лабораторная работа №1

Дидусь К.В. Группа НКНбд-01-18

31.03.2021

Прагматика выполнения лабораторной работы

- Git невероятно полезный и популярный инструмент
- Стандартное решения для контроля версий

Научиться создавать репозитории в github, использовать Git Bash, загружать файлы и папки на GitHub с помощью Git Bash.

- Создать репозиторий
- Первичная настройка Git Bash
- Создать git-репозиторий из каталога на компьютере
- Создать каталог и файл и дать commit
- Загрузить на GitHub

Выполнение лабораторной работы

Создание репозитория

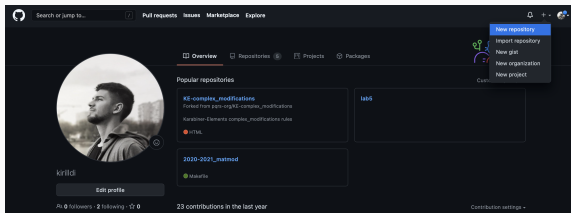



Рис. 1: Создание репозитория

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner *


Repository name *

 kirilIdi ▾


/ test_repo ✓

Great repository names are short and memorable. Need inspiration? How about **furry-computing-machine?**

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.

Рис. 2: Создание репозитория

Первичная настройка Git Bash

```
((base) MacBook-Pro-Kirill:~ kirilldi$ git config --global core.autocrlf input ]  
((base) MacBook-Pro-Kirill:~ kirilldi$ git config --global core.safecrlf true ]  
((base) MacBook-Pro-Kirill:~ kirilldi$ git config --global core.quotepath off ]  
((base) MacBook-Pro-Kirill:~ kirilldi$
```

Рис. 3: Команды первоначальной настройки

Создание git-репозитория из каталога на компьютере.

```
(base) MacBook-Pro-Kirill:~ kirilldi$ cd test_dir/
(base) MacBook-Pro-Kirill:test_dir kirilldi$ git init
Инициализирован пустой репозиторий Git в /Users/kirilldi/test_dir/.git/
(base) MacBook-Pro-Kirill:test_dir kirilldi$ touch README.md
(base) MacBook-Pro-Kirill:test_dir kirilldi$ echo "test note">README.md
(base) MacBook-Pro-Kirill:test_dir kirilldi$ git add .
(base) MacBook-Pro-Kirill:test_dir kirilldi$ git commit -m 'feat(main): add README'
[master (корневой коммит) e6f8e66] feat(main): add README
1 file changed, 1 insertion(+)
create mode 100644 README.md
(base) MacBook-Pro-Kirill:test_dir kirilldi$ git remote add origin https://github.com/kirilldi/test_repo.git
(base) MacBook-Pro-Kirill:test_dir kirilldi$ git branch -M main
(base) MacBook-Pro-Kirill:test_dir kirilldi$ git push -u origin main
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 232 bytes | 232.00 KiB/s, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/kirilldi/test_repo.git
 * (new branch)    main -> main
Ветка «main» отслеживает внешнюю ветку «main» из «origin».
(base) MacBook-Pro-Kirill:test_dir kirilldi$
```

Рис. 4: Создание git-репозитория

Создание каталога и файла

```
((base) MacBook-Pro-Kirill:test_dir kirilldi$ mkdir test_lab01
((base) MacBook-Pro-Kirill:test_dir kirilldi$ cd test_lab01/
((base) MacBook-Pro-Kirill:test_lab01 kirilldi$ touch hello.md
((base) MacBook-Pro-Kirill:test_lab01 kirilldi$ echo "Hello,world!">hello.md
-bash: !": event not found
((base) MacBook-Pro-Kirill:test_lab01 kirilldi$ echo "Hello, world">hello.md
((base) MacBook-Pro-Kirill:test_lab01 kirilldi$ git add hello.md
((base) MacBook-Pro-Kirill:test_lab01 kirilldi$ git commit -m "feat(main):add hello"
[main d2f6914] feat(main):add hello
1 file changed, 1 insertion(+)
 create mode 100644 test_lab01/hello.md
((base) MacBook-Pro-Kirill:test_lab01 kirilldi$ git push
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (4/4), 341 bytes | 341.00 KiB/s, готово.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/kirilldi/test_repo.git
 e6f8e66..d2f6914  main -> main
```

Рис. 5: Создание каталога и файла

Загрузка на GitHub

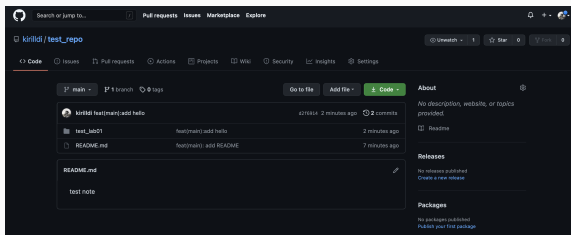


Рис. 6: Репозиторий на GitHub

Научился создавать репозитории в github,
использовать Git Bash, загружать файлы и папки на
GitHub с помощью Git Bash.