

# Презентация по лабораторной работе №4

---

Шубина С.А.

04 марта 2024

Российский университет дружбы народов, Москва, Россия

# Информация

---

- Шубина София Антоновна
- Студентка, НПИбд-02-23
- факультет физико-математических и естественных наук, направление прикладная информатиика
- Российский университет дружбы народов
- 1132232885@pfur.ru

Получение навыков правильной работы с репозиториями git.

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

Рабочий процесс Gitflow Workflow. Будем описывать его с использованием пакета git-flow







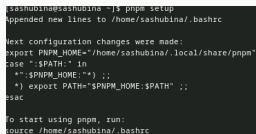
## Установка Node.js

На Node.js базируется программное обеспечение для семантического версионирования и общепринятых коммитов. Fedora `dnf install nodejs`  
apt-get install pnpm

[illegible]

### Рис. 2: Установка Node.js

Для работы с Node.js добавим каталог с исполняемыми файлами, устанавливаемыми yarn, в переменную PATH. Запустим: `pnpm setup`



```
sashubina@sashubina ~$ pnpm setup
Appended new lines to /home/sashubina/.bashrc

Next configuration changes were made:
export PNPM_HOME="/home/sashubina/.local/share/pnpm"
case "$PATH:" in
  *"$PNPM_HOME:") ;;
  *) export PATH="$PNPM_HOME:$PATH" ;;
esac

To start using pnpm, run:
source /home/sashubina/.bashrc
```

**Рис. 3:** Запуск `pnpm setup`

## Перелогинемся или выполним:

`source ~/.bashrc` Общепринятые коммиты `commitizen` Данная программа используется для помощи в форматировании коммитов. `pnpm add -g commitizen` При этом устанавливается скрипт `git-cz`, который мы и будем использовать для коммитов.

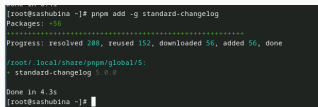
```
root@archlinux: ~# pnpm add -g commitizen

Update available! 4.12.0 < 4.15.4.
Changelog: https://github.com/pnpm/pnpm/releases/tag/v4.15.4
Run "pnpm add -g pnpm" to update.
Follow @pnpmjs for updates: https://twitter.com/pnpmjs

Packages: 152
Progress: resolved 152, reused 0, downloaded 152, added 152, done
root/.local/share/pnpm/global/5:
commitizen ~ 4.12.0
Done in 6.4s
root@archlinux: ~#
```

Рис. 4: Установка программы и скрипта

Данная программа используется для помощи в создании логов. `pnpm add -g standard-changelog`



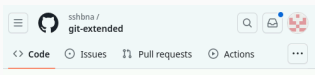
```
[root@sashubina ~]# pnpm add -g standard-changelog
Packages: +56
Progress: resolved 208, reused 152, downloaded 56, added 56, done

/root/.local/share/pnpm/global/5:
standard-changelog : 0.0

Done in 4.3s
[root@sashubina ~]#
```

Рис. 5: Скачивание standard-changelog

Создание репозитория git Подключение репозитория к github Создадим репозиторий на GitHub. Для примера назовём его git-extended.



**Рис. 6:** Создание репозитория

# Делаем первый коммит и выкладываем на github:

Для начала клонируем репозиторий

```
[sashubina@sashubina ~]$ git clone --recursive git@github.com:sshbn/git-extended.git
Клонирование в «git-extended»...
```

**Рис. 7:** Клонировем репозиторий

# git commit -m "first commit"

git remote add origin git@github.com:/git-extended.git git push -u origin master

```
[sashubina@sashubina git-extended]$ git commit -m "first commit"
[main b804235] first commit
1 file changed, 1 insertion(+)
[sashubina@sashubina git-extended]$ git remote add origin git@github.com:
sshbna/git-extended.git
error: внешний репозиторий origin уже существует
[sashubina@sashubina git-extended]$ git push -u origin master
error: src refspec master ничему не соответствует
error: не удалось отправить некоторые ссылки в «github.com:sshbna/git-extended.git»
[sashubina@sashubina git-extended]$ git push -u origin main
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (6/6), 1.71 КиБ | 1.71 МБ/с, готово.
Всего 6 (изменений 0), повторно использовано 0 (изменений 0), повторно и
спользовано пакетов 0
To github.com:sshbna/git-extended.git
 * [new branch]    main -> main
branch 'main' set up to track 'origin/main'.
[sashubina@sashubina git-extended]$
```

Рис. 8: Делаем первый commit

## Конфигурация для пакетов Node.js pnpm init

```
{sashubina@sashubina git-extended}$ pnpm init
Wrote to /home/sashubina/git-extended/package.json

{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\Error: no test specified\\ && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
{sashubina@sashubina git-extended}$
```

**Рис. 9:** Конфигурация для пакетов Node.js



## Необходимо заполнить несколько параметров пакета.

Название пакета. Лицензия пакета. Список лицензий для npm: <https://spdx.org/licenses/>. Предлагается выбирать лицензию CC-BY-4.0. Сконфигурируем формат коммитов. Для этого добавим в файл package.json команду для формирования коммитов: “config”: { “commitizen”: { “path”: “cz-conventional-changelog” } } ## Таким образом, файл package.json приобретает вид: { “name”: “git-extended”, “version”: “1.0.0”, “description”: “Git repo for educational purposes”, “main”: “index.js”, “repository”: “git@github.com:username/git-extended.git”, “author”: “Name Surname username@gmail.com”, “license”: “CC-BY-4.0”, “config”: { “commitizen”: { “path”: “cz-conventional-changelog” } } }

Вносим изменения

**Рис. 10:** Вносим изменения

## Добавим новые файлы:

git add . Выполним коммит: git cz Отправим на github: git push

```
foot
  'author': '',
  'license': 'ISC'
}
[sashubina@sashubina git-extended]$ nano package.json
[sashubina@sashubina git-extended]$ git add .
[sashubina@sashubina git-extended]$ git cz
Parsing JSON at /home/sashubina/git-extended/package.json for commitizen
config failed:

[sashubina@sashubina git-extended]$ nano ~/.czrc
[sashubina@sashubina git-extended]$ nano package.json
[sashubina@sashubina git-extended]$ git cz
Parsing JSON at /home/sashubina/git-extended/package.json for commitizen
config failed:

[sashubina@sashubina git-extended]$ nano package.json
[sashubina@sashubina git-extended]$ git cz
cz-cli@4.3.0, cz-conventional-changelog@3.3.0

? Select the type of change that you're committing: feat:      A new
feature
? What is the scope of this change (e.g. component or file name): (press
enter to skip) readme.md
? Write a short, imperative tense description of the change (max 83
chars):
? Provide a longer description of the change: (press enter to skip)

? Are there any breaking changes? No
? Does this change affect any open issues? No
[main baacc87] feat(readme.md): added file
1 file changed, 15 insertions(+)
create mode 100644 package.json
[sashubina@sashubina git-extended]$ git push
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
При сканировании используется до 6 потоков
Сканирование объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 1.17 KiB | 170.00 KiB/c, готово.
Всего 3 (изменения 0), повторно использовано 0 (изменения 0), повторно и
использовано пакетов 0
To github.com:sashubina/git-extended.git
bb04235..baacc87  main -> main
[sashubina@sashubina git-extended]$
```

Рис. 11: Добавим новые файлы

Инициализируем git-flow `git flow init` Префикс для ярлыков установим в v.

```
To force reinitialization, use: git flow init -f
[sashubina@sashubina git-extended]$ git flow init -f

Which branch should be used for bringing forth production releases?
- develop
- main
Branch name for production releases: [main]

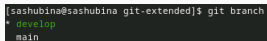
Which branch should be used for integration of the "next release"?
- develop
Branch name for "next release" development: [develop]

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? [/home/sashubina/git-extended/.git/hooks]
[sashubina@sashubina git-extended]$
```

Рис. 12: Инициализируем git-flow

# Проверьте, что Вы на ветке develop:

git branch

A terminal window with a dark background. The prompt is [sashubina@sashubina git-extended]\$ and the command is git branch. The output shows two branches: \* develop (highlighted in green) and main.

```
[sashubina@sashubina git-extended]$ git branch
* develop
main
```

**Рис. 13:** Проверим, что мы на ветке

# Загрузите весь репозиторий в хранилище:

`git push --all`

```
[sashubina@sashubina git-extended]$ git push --all
Всего 0 (изменений 0), повторно использовано 0 (изменений 0), повторно и
использовано пакетов 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/sashbna/git-extended/pull/new/develop
remote:
To github.com:sashbna/git-extended.git
 * [new branch]    develop -> develop
[sashubina@sashubina git-extended]$
```

Рис. 14: Загрузим весь репозиторий в хранилище

## Установим внешнюю ветку как вышестоящую для этой ветки:

git branch --set-upstream-to=origin/develop develop Создадим релиз с версией 1.0.0 git flow release start 1.0.0

```
[sashubina@sashubina git-extended]$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
[sashubina@sashubina git-extended]$ git flow release start 1.0.0
Переключились на новую ветку «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'

[sashubina@sashubina git-extended]$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
[sashubina@sashubina git-extended]$ git flow release start 1.0.0
Fatal: There is an existing release branch '1.0.0'. Finish that one first.
[sashubina@sashubina git-extended]$
```

Рис. 15: Установим внешнюю ветку как вышестоящую для этой ветки

# Создадим журнал изменений

standard-changelog --first-release Добавим журнал изменений в индекс git add CHANGELOG.md git commit -am 'chore(site): add changelog' {.column width="30%"}

```
[sashubina@sashubina git-extended]$ standard-changelog
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
[sashubina@sashubina git-extended]$ standard-changelog --first-release
✓ output changes to CHANGELOG.md
[sashubina@sashubina git-extended]$ git add CHANGELOG.md
[sashubina@sashubina git-extended]$ git commit -am 'chore(site):update c
hangelog'
[release/1.0.0 6f42435] chore(site):update changelog
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
[sashubina@sashubina git-extended]$
```

**Рис. 16:** Создадим журнал изменений и добавим его в индекс

# Зальём релизную ветку в основную ветку

git flow release finish 1.0.0 Отправим данные на github git push --all git push --tags

```
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
fatal: нет описания метки?
fatal: Tagging failed. Please run finish again to retry.
[sashubina@sashubina git-extended]$ git flow release finish 1.0.0
Branches 'main' and 'origin/main' have diverged.
and local branch 'main' is ahead of 'origin/main'.
Кем на «main»
Ваша ветка опережает «origin/main» на 2 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
Переключились на ветку «develop»
Эта ветка соответствует «origin/develop».
Merge made by the 'ort' strategy.
CHANGELOG.md | 9 ++++++++
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
Ветка release/1.0.0 удалена (была 6f42435).

Summary of actions:
- Release branch 'release/1.0.0' has been merged into 'main'
- The release was tagged 'v1.0.0'
- Release tag 'v1.0.0' has been back-merged into 'develop'
- Release branch 'release/1.0.0' has been locally deleted
- You are now on branch 'develop'

[sashubina@sashubina git-extended]$ git push --all
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
При скачивании изменений используется до 6 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 2.80 КиБ | 2.80 МБ/с, готово.
Всего 5 (изменений 0), повторно использовано 0 (изменений 0), повторно и
использовано пакетов 0
To github.com:sshbna/git-extended.git
baacc87..ff941f4 develop -> develop
baacc87..d3e008a main -> main
[sashubina@sashubina git-extended]$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 157 байтов | 157.00 КиБ/с, готово.
Всего 1 (изменений 0), повторно использовано 0 (изменений 0), повторно и
использовано пакетов 0
To github.com:sshbna/git-extended.git
* [new tag] v1.0.0 -> v1.0.0
[sashubina@sashubina git-extended]$
```



## Создадим релиз на github. Для этого будем использовать утилиты работы с github:

`gh release create v1.0.0 -F CHANGELOG.md` Работа с репозитором git  
Разработка новой функциональности Создадим ветку для новой функциональности: `git flow feature start feature_branch` Далее, продолжаем работу с git как обычно. По окончании разработки новой функциональности следующим шагом следует объединить ветку `feature_branch` с `develop`: `git flow feature finish feature_branch`

Создание релиза git-flow Создадим релиз с версией 1.2.3: `git flow release start 1.2.3` Обновите номер версии в файле `package.json`. Установите её в 1.2.3.

```
GNU nano 7.2 package.json Изменён
{
  "name": "git-extended",
  "version": "1.2.3",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "keywords": [],
  "repository": "git@github.com:sshbna/git-extended.git",
  "author": "Shubina Sofia 1132232885@pfur.ru",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

## Создадим журнал изменений

standard-changelog Добавим журнал изменений в индекс git add  
CHANGELOG.md git commit -am 'chore(site): update changelog' Зальём  
релизную ветку в основную ветку git flow release finish 1.2.3

# Отправим данные на github

git push --all git push --tags Создадим релиз на github с комментарием из журнала изменений: gh release create v1.2.3 -F CHANGELOG.md

```
[sashubina@sashubina git-extended]$ git flow release finish 1.2.3
Fatal: Branch 'release/1.2.3' does not exist and is required.
[sashubina@sashubina git-extended]$ git push --all
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При скатии изменений используется до 6 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 2.77 КиБ | 2.77 МБ/с, готово.
Всего 6 (изменений 2), повторно использовано 0 (изменений 0), повторно и
спользовано пакетов 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:sshbn/git-extended.git
ff941f4..b768612 develop -> develop
d3e008a..0b5b270 main -> main
[sashubina@sashubina git-extended]$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 158 байтов | 158.00 КиБ/с, готово.
Всего 1 (изменений 0), повторно использовано 0 (изменений 0), повторно и
спользовано пакетов 0
To github.com:sshbn/git-extended.git
* [new tag] v1.2.3 -> v1.2.3
[sashubina@sashubina git-extended]$ gh release create v1.2.3 -F CHANGELO
G.md
https://github.com/sshbn/git-extended/releases/tag/v1.2.3
[sashubina@sashubina git-extended]$
```

**Рис. 19:** Создание журнала изменений, добавление в индекс, заливаем ветку в основную и отправляем данные

## **Выводы**

---

Я получила правильные навыки работы с репозиториями git.