

Лабораторная работа №4

Продвинутое использование Git

Губайдуллина Софья Романовна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	10
5	Выводы	23
	Список литературы	24

Список иллюстраций

4.1	Установка репозитория Copr	10
4.2	Установка git flow	11
4.3	Установка Node.js	11
4.4	Настройка Node.js	12
4.5	Pnpm setup	12
4.6	Установка commitizen	13
4.7	Установка standard-changelog	13
4.8	Создание нового репозитория	14
4.9	Первый коммит	14
4.10	npm init	14
4.11	Заполнение параметров пакета	15
4.12	Git add	15
4.13	Git cz	16
4.14	Отправка на сервер	16
4.15	Git flow init	16
4.16	Git branch и загрузка всего репозитория	17
4.17	Загрузка всего репозитория	17
4.18	Установка внешней ветки	17
4.19	Версия 1.0.0	18
4.20	Журнал изменений	18
4.21	Commit changes	18
4.22	Перемещение релизной ветки в основную	19
4.23	Отправка данных на GitHub	19
4.24	Релиз 1.0.0	19
4.25	Создание ветки новой функциональности	20
4.26	Создание релиза 1.2.3	20
4.27	Создание журнала изменений	20
4.28	Добавление в индекс	21
4.29	Перемещение ветки и отправка на сервер	21
4.30	Отправка на сервер	21
4.31	Создание релиза с комментарием	22

Список таблиц

1 Цель работы

Получение навыков правильной работы с репозиториями git.

2 Задание

1. Установка Gitflow;
2. Установка Node.js;
3. Настройка Node.js;
4. Общепринятые коммиты;
5. Практический сценарий использования git;
6. Работа с репозиторием git.

3 Теоретическое введение

1) Процесс работы с Gitflow Основные ветки (master) и ветки разработки (develop)

Для фиксации истории проекта в рамках этого процесса вместо одной ветки master используются две ветки. В ветке master хранится официальная история релиза, а ветка develop предназначена для объединения всех функций. Кроме того, для удобства рекомендуется присваивать всем коммитам в ветке master номер версии. При использовании библиотеки расширений git-flow нужно инициализировать структуру в существующем репозитории:

```
git flow init
```

2) Функциональные ветки (feature)

Под каждую новую функцию должна быть отведена собственная ветка, которую можно отправлять в центральный репозиторий для создания резервной копии или совместной работы команды. Ветки feature создаются не на основе master, а на основе develop. Когда работа над функцией завершается, соответствующая ветка сливается обратно с веткой develop. Функции не следует отправлять напрямую в ветку master. Как правило, ветки feature создаются на основе последней ветки develop.

3) Ветки выпуска (release)

Когда в ветке develop оказывается достаточно функций для выпуска, из ветки develop создаётся ветка release. Создание этой ветки запускает следующий цикл

выпуска, и с этого момента новые функции добавить больше нельзя — допускается лишь отладка, создание документации и решение других задач. Когда подготовка релиза завершается, ветка `release` сливается с `master` и ей присваивается номер версии. После нужно выполнить слияние с веткой `develop`, в которой с момента создания ветки релиза могли возникнуть изменения. Благодаря тому, что для подготовки выпусков используется специальная ветка, одна команда может дорабатывать текущий выпуск, в то время как другая команда продолжает работу над функциями для следующего.

4) Ветки исправления (`hotfix`)

Ветки поддержки или ветки `hotfix` используются для быстрого внесения исправлений в рабочие релизы. Они создаются от ветки `master`. Это единственная ветка, которая должна быть создана непосредственно от `master`. Как только исправление завершено, ветку следует объединить с `master` и `develop`. Ветка `master` должна быть помечена обновлённым номером версии. Наличие специальной ветки для исправления ошибок позволяет команде решать проблемы, не прерывая остальную часть рабочего процесса и не ожидая следующего цикла релиза.

Семантическое версионирование описывается в манифесте семантического версионирования. Кратко его можно описать следующим образом:

Версия задаётся в виде кортежа `МАЖОРНАЯ_ВЕРСИЯ.МИНОРНАЯ_ВЕРСИЯ.ПАТЧ`. Номер версии следует увеличивать: `МАЖОРНУЮ` версию, когда сделаны обратно несовместимые изменения `API`. `МИНОРНУЮ` версию, когда вы добавляете новую функциональность, не нарушая обратной совместимости. `ПАТЧ`-версию, когда вы делаете обратно совместимые исправления. Дополнительные обозначения для предрелизных и билд-метаданных возможны как дополнения к `МАЖОРНАЯ.МИНОРНАЯ.ПАТЧ` формату.

Коммиты. Базовые типы коммитов

`fix`: — коммит типа `fix` исправляет ошибку (`bug`) в вашем коде (он соответствует `PATCH` в `SemVer`). `feat`: — коммит типа `feat` добавляет новую функцию

(feature) в ваш код (он соответствует MINOR в SemVer). BREAKING CHANGE: — коммит, который содержит текст BREAKING CHANGE: в начале своего не обязательного тела сообщения (body) или в подвале (footer), добавляет изменения, нарушающие обратную совместимость вашего API (он соответствует MAJOR в SemVer). revert: — если фиксация отменяет предыдущую фиксацию. Другое: коммиты с типами, которые отличаются от fix: и feat:, также разрешены. Например, **[commitlint/config-conventional?]** (основанный на The Angular convention) рекомендует: chore:, docs:, style:, refactor:, perf:, test:, и другие.

4 Выполнение лабораторной работы

1. Для того, чтобы установить git flow, мне необходимо установить коллекцию репозитория Copr (рис. 4.1).

```
srgubayjdullina@srgubayjdullina:~$ sudo dnf copr enable elegos/gitflow
[sudo] пароль для srgubayjdullina:
Включение репозитория Copr. Обратите внимание, что этот репозиторий
не является частью основного дистрибутива, и качество может отличаться.

Проект Fedora не имеет какого-либо влияния на содержимое этого
репозитория за рамки правил, описанных в Вопросах и Ответах Copr в
<https://docs.pagure.org/copr.copr/user\_documentation.html#what-i-can-build-in-copr>,
а качество и безопасность пакетов не поддерживаются на каком-либо уровне.

Не отправляйте сообщения об ошибках этих пакетов в Fedora
Bugzilla. В случае возникновения проблем обращайтесь к владельцу этого репозитория.

Do you really want to enable copr.fedorainfracloud.org/elegos/gitflow? [y/N]:
```

Рис. 4.1: Установка репозитория Copr

Далее устанавливаю сам git flow через консоль (рис. 4.2)

```
srgubayjdullina@srgubayjdullina:~ — sudo dnf install gitflow
Do you really want to enable copr.fedorainfracloud.org/elegos/gitflow? [y/N]: y
Репозиторий успешно подключен.
srgubayjdullina@srgubayjdullina:~$ sudo dnf install gitflow
Copr repo for gitflow owned by elegos      1.2 kB/s | 1.5 kB      00:01
Последняя проверка окончания срока действия метаданных: 0:00:01 назад, Пн 04 мар
2024 19:11:02.
Зависимости разрешены.
=====
Пакет  Архитектура      Версия      Репозиторий      Размер
=====
Установка:
gitflow
x86_64 1.12.3-1.fc34 copr:copr.fedorainfracloud.org:elegos:gitflow 57 k
Результат транзакции
=====
Установка 1 Пакет

Объем загрузки: 57 k
Объем изменений: 262 k
Продолжить? [д/н]:
```

Рис. 4.2: Установка git flow

2. Так же через терминал устанавливаю Node.js - на нем базируется программное обеспечение для семантического версионирования и общепринятых коммитов (рис. 4.3)

```
srgubayjdullina@srgubayjdullina:~$ sudo dnf install nodejs
Последняя проверка окончания срока действия метаданных: 0:01:57 назад, Пн 04 мар
2024 19:11:02.
Зависимости разрешены.
=====
Пакет      Архитектура      Версия      Репозиторий      Размер
=====
Установка:
nodejs     x86_64           1:20.10.0-3.fc39 updates          48 k
Установка зависимостей:
nodejs-libs x86_64           1:20.10.0-3.fc39 updates          15 M
Установка слабых зависимостей:
nodejs-docs noarch           1:20.10.0-3.fc39 updates          8.1 M
nodejs-full-i18n x86_64           1:20.10.0-3.fc39 updates          8.5 M
nodejs-npm   x86_64           1:10.2.3-1.20.10.0.3.fc39 updates          2.2 M
Результат транзакции
=====
Установка 5 Пакетов

Объем загрузки: 34 М
```

Рис. 4.3: Установка Node.js

3. Для работы с Node.js добавляю каталог с исполняемыми файлами, установ-

ливаемыми yarn, в переменную PATH (рис. 4.4).

```
srgubayjdullina@srgubayjdullina:~ — sudo dnf install pnpm
sudo: apt-get: команда не найдена
srgubayjdullina@srgubayjdullina:~$ sudo dnf install pnpm
Последняя проверка окончания срока действия метаданных: 0:03:35 назад, Пн 04 мар
2024 19:11:02.
Зависимости разрешены.
=====
Пакет      Архитектура  Версия      Репозиторий  Размер
=====
Установка:
pnpm       noarch       8.12.0-1.fc39 updates      2.6 М
=====
Результат транзакции
=====
Установка 1 Пакет

Объем загрузки: 2.6 М
Объем изменений: 13 М
Продолжить? [д/н]: д
Загрузка пакетов:
pnpm-8.12.0-1.fc39.noarch.rpm      4.6 MB/s | 2.6 MB  00:00
=====
Общий размер      2.7 MB/s | 2.6 MB  00:00
Проверка транзакции
```

Рис. 4.4: Настройка Node.js

Запускаю pnpm setup, перелогинюсь и захожу (рис. 4.5).

```
srgubayjdullina@srgubayjdullina:~
Тест транзакции проведен успешно.
Выполнение транзакции
Подготовка      : 1/1
Установка       : pnpm-8.12.0-1.fc39.noarch 1/1
Запуск скрипта  : pnpm-8.12.0-1.fc39.noarch 1/1
Проверка        : pnpm-8.12.0-1.fc39.noarch 1/1

Установлен:
pnpm-8.12.0-1.fc39.noarch

Выполнено!
srgubayjdullina@srgubayjdullina:~$ pnpm setup
Appended new lines to /home/srgubayjdullina/.bashrc

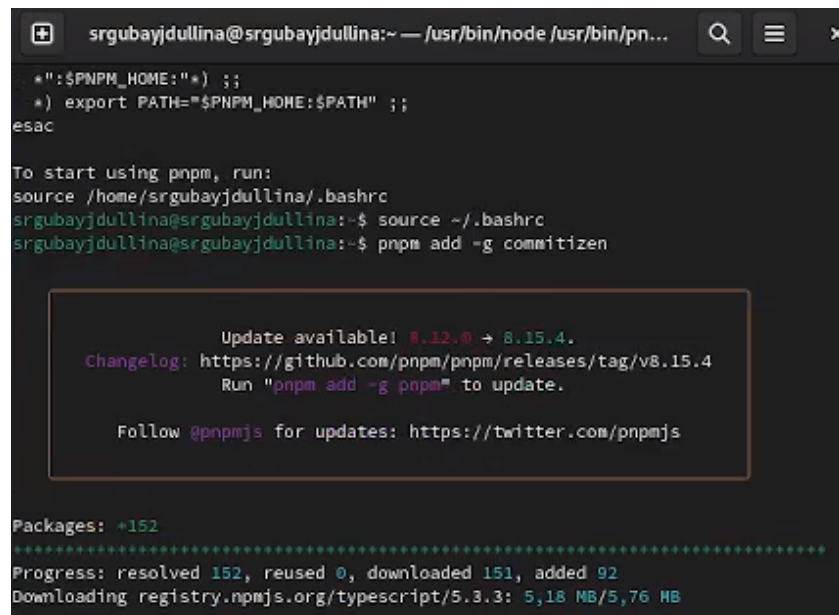
Next configuration changes were made:
export PNPM_HOME="/home/srgubayjdullina/.local/share/pnpm"
case ":$PATH:" in
  *) export PATH="$PNPM_HOME:$PATH" ;;
  *) export PATH="$PNPM_HOME:$PATH" ;;
esac

To start using pnpm, run:
source /home/srgubayjdullina/.bashrc
srgubayjdullina@srgubayjdullina:~$
```

Рис. 4.5: Pnpm setup

4. Устанавливаю программу commitizen, которая используется для помощи в

форматировании коммитов. При этом устанавливается скрипт git-cz, который мы и будем использовать для коммитов (рис. 4.6).



```
srgubayjdullina@srgubayjdullina:~ — /usr/bin/node /usr/bin/pn...
*):$PNPM_HOME:*) ;;
*) export PATH="$PNPM_HOME:$PATH" ;;
esac

To start using pnpm, run:
source /home/srgubayjdullina/.bashrc
srgubayjdullina@srgubayjdullina:~$ source ~/.bashrc
srgubayjdullina@srgubayjdullina:~$ pnpm add -g commitizen

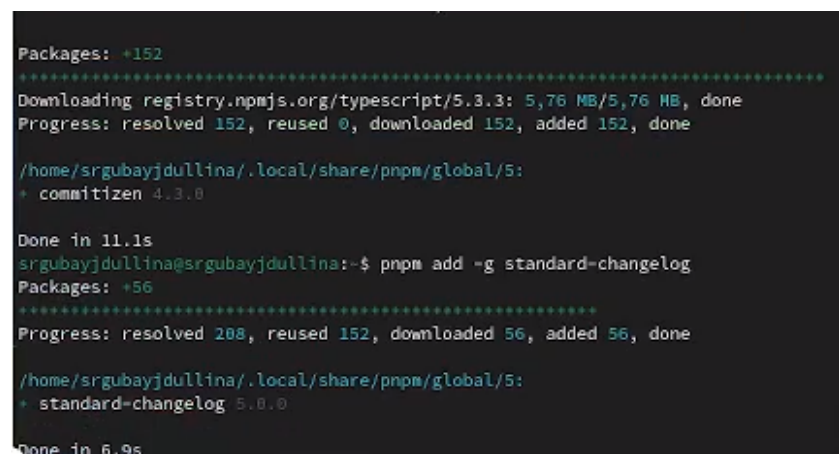
Update available! 8.12.0 → 8.15.4.
Changelog: https://github.com/pnpm/pnpm/releases/tag/v8.15.4
Run "pnpm add -g pnpm" to update.

Follow @pnpmjs for updates: https://twitter.com/pnpmjs

Packages: +152
Progress: resolved 152, reused 0, downloaded 151, added 92
Downloading registry.npmjs.org/typescript/5.3.3: 5.18 MB/5.76 MB
```

Рис. 4.6: Установка commitizen

Устанавливаю программу standard-changelog, которая используется для помощи в создании логов (рис. 4.7).



```
Packages: +152
Progress: resolved 152, reused 0, downloaded 152, added 152, done

/home/srgubayjdullina/.local/share/pnpm/global/5:
+ commitizen 4.3.0

Done in 11.1s
srgubayjdullina@srgubayjdullina:~$ pnpm add -g standard-changelog
Packages: +56
Progress: resolved 208, reused 152, downloaded 56, added 56, done

/home/srgubayjdullina/.local/share/pnpm/global/5:
+ standard-changelog 5.0.0

Done in 6.9s
```

Рис. 4.7: Установка standard-changelog

5. Созданию новый репозиторий git и назову его git-extended (рис. 4.8). При помощи git commit делаю первый коммит и выкладываю на гитхаб (рис.

4.9).

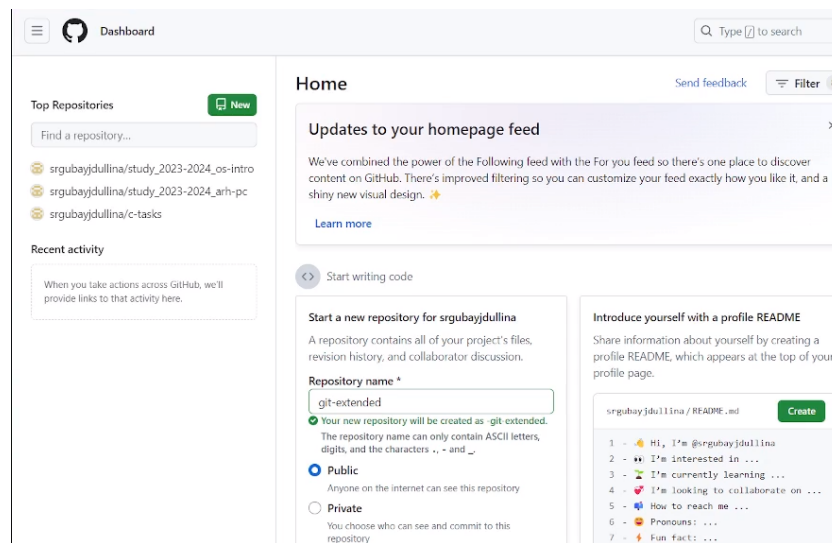


Рис. 4.8: Создание нового репозитория

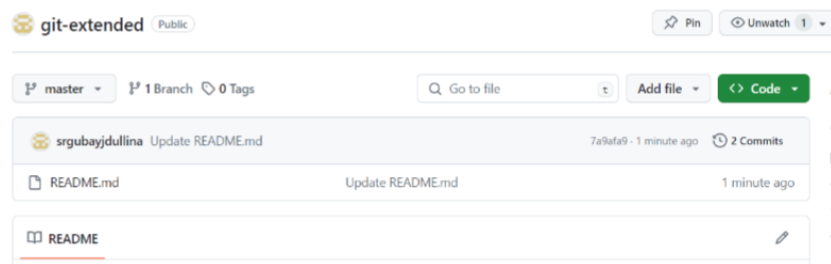


Рис. 4.9: Первый коммит

Далее требуется конфигурация для пакетов Node.js (рис. 4.10).

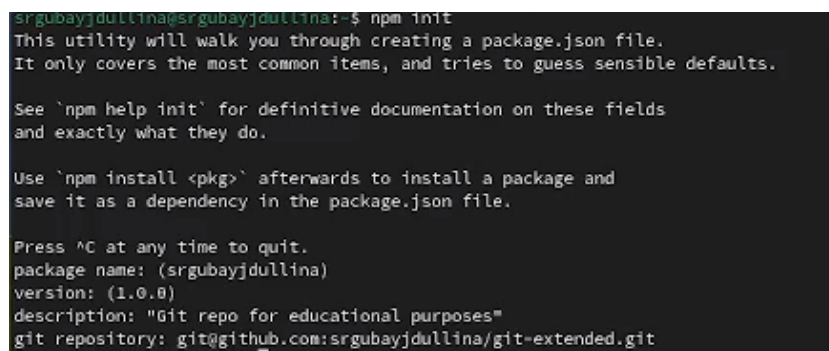
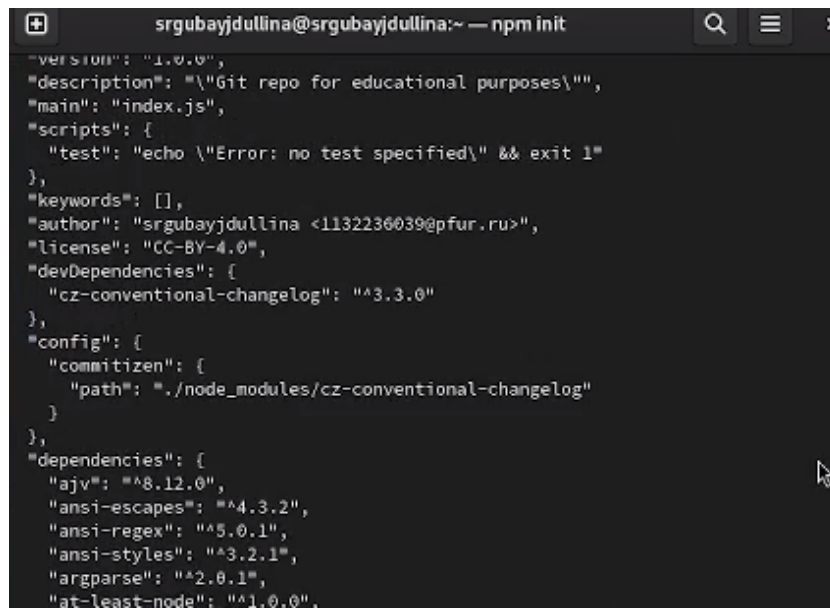


Рис. 4.10: nmp init

Требуется заполнить некоторые параметры для пакетов Node.js (рис. 4.11).

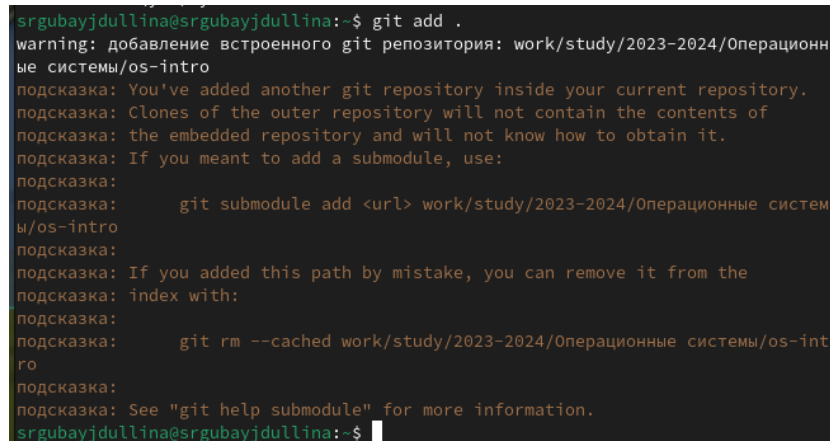


```
+ srgubayjdullina@srgubayjdullina:~ — npm init
{
  "version": "1.0.0",
  "description": "\"Git repo for educational purposes\"",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [],
  "author": "srgubayjdullina <1132236039@pfur.ru>",
  "license": "CC-BY-4.0",
  "devDependencies": {
    "cz-conventional-changelog": "^3.3.0"
  },
  "config": {
    "commitizen": {
      "path": "./node_modules/cz-conventional-changelog"
    }
  },
  "dependencies": {
    "ajv": "^8.12.0",
    "ansi-escapes": "^4.3.2",
    "ansi-regex": "^5.0.1",
    "ansi-styles": "^3.2.1",
    "argparse": "^2.0.1",
    "at-least-node": "^1.0.0",

```

Рис. 4.11: Заполнение параметров пакета

Заканчиваем добавлением новых файлов, выполнением коммитов и отправкой файлов на сервер (рис. 4.12).



```
srgubayjdullina@srgubayjdullina:~$ git add .
warning: добавление встроенного git репозитория: work/study/2023-2024/Операционн
ые системы/os-intro
подсказка: You've added another git repository inside your current repository.
подсказка: Clones of the outer repository will not contain the contents of
подсказка: the embedded repository and will not know how to obtain it.
подсказка: If you meant to add a submodule, use:
подсказка:
подсказка:     git submodule add <url> work/study/2023-2024/Операционные систем
ы/os-intro
подсказка:
подсказка: If you added this path by mistake, you can remove it from the
подсказка: index with:
подсказка:
подсказка:     git rm --cached work/study/2023-2024/Операционные системы/os-int
ro
подсказка:
подсказка: See "git help submodule" for more information.
srgubayjdullina@srgubayjdullina:~$
```

Рис. 4.12: Git add

Выполненный коммит (рис. 4.13):

```
srgubayjdullina@srgubayjdullina:~$ 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)
? Write a short, imperative tense description of the change (max 94 chars):
  (12) git-extended
? Provide a longer description of the change: (press enter to skip)

? Are there any breaking changes? Yes
? A BREAKING CHANGE commit requires a body. Please enter a longer description of the commit itself:
-
? Describe the breaking changes:

? Does this change affect any open issues? No
Автоматическая упаковка репозитория в фоне, для оптимальной производительности.
Смотрите «git help gc» руководства по ручной очистке.
[master 2fc9f75] feat: git-extended
```

Рис. 4.13: Git cz

Git push (рис. 4.14):

```
srgubayjdullina@srgubayjdullina:~$ git push
Перечисление объектов: 16988, готово.
Подсчет объектов: 100% (16988/16988), готово.
Сжатие объектов: 100% (11340/11340), готово.
Запись объектов: 100% (16986/16986), 541.66 МиБ | 2.06 МиБ/с, готово.
Всего 16986 (изменений 4237), повторно использовано 16962 (изменений 4225), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4237/4237), done.
```

Рис. 4.14: Отправка на сервер

Теперь нам нужно инициализировать git flow (рис. 4.15).

```
srgubayjdullina@srgubayjdullina:~$ git flow init

Which branch should be used for bringing forth production releases?
- master
Branch name for production releases: [master] master
Branch name for "next release" development: [develop] develop

How to name your supporting branch prefixes?
Feature branches? [feature/] feature
Bugfix branches? [bugfix/] bugfix
Release branches? [release/] release
Hotfix branches? [hotfix/] hotfix
Support branches? [support/] support
Version tag prefix? []
Hooks and filters directory? [/home/srgubayjdullina/.git/hooks]
```

Рис. 4.15: Git flow init

Проверяю, что я нахожусь на ветке develop и загружаю весь репозиторий в хранилище (рис. 4.16).


```

How to name your supporting branch prefixes?
Feature branches? [feature/] feature
Bugfix branches? [bugfix/] bugfix
Release branches? [release/] release
Hotfix branches? [hotfix/] hotfix
Support branches? [support/] support
Version tag prefix? []
Hooks and filters directory? [/home/srgubayjdullina/.git/hooks]
srgubayjdullina@srgubayjdullina:~$ git branch
* develop
  master
srgubayjdullina@srgubayjdullina:~$ git push --all
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names.

```

Рис. 4.16: Git branch и загрузка всего репозитория

Git push –all (рис. 4.17):

```

srgubayjdullina@srgubayjdullina:~$ git push --all
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhbpZisF/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.
Всего 0 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: This repository moved. Please use the new location:
remote:   git@github.com:srgubayjdullina/git-extended.git
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/srgubayjdullina/git-extended/pull/new/develop
remote:
To github.com:srgubayjdullina/~git-extended.git
 * [new branch]      develop -> develop

```

Рис. 4.17: Загрузка всего репозитория

Устанавливаю внешнюю ветку как вышестоящую для этой ветки (рис. 4.18).

```

srgubayjdullina@srgubayjdullina:~$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.

```

Рис. 4.18: Установка внешней ветки

Создаю релиз с версией 1.0.0 (рис. 4.19).

```

srgubayjdullina@srgubayjdullina:~$ git flow release start 1.0.0
Переключились на новую ветку «release1.0.0»

Summary of actions:
- A new branch 'release1.0.0' was created, based on 'develop'
- You are now on branch 'release1.0.0'

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'

```

Рис. 4.19: Версия 1.0.0

Создаю журнал изменений git standard, а так же добавляю его в индекс (рис. 4.20).

```

[sudo] пароль для srgubayjdullina:
added 71 packages in 17s

16 packages are looking for funding
  run `npm fund` for details
npm notice
npm notice New minor version of npm available! 10.2.3 -> 10.5.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.5.0
npm notice Run `npm install -g npm@10.5.0` to update!
npm notice
srgubayjdullina@srgubayjdullina:~$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md

```

Рис. 4.20: Журнал изменений

Commit (рис. 4.21):

```

srgubayjdullina@srgubayjdullina:~$ git commit -am 'chore(site): add changelog'
[release1.0.0 1a91dc8a] chore(site): add changelog
1 file changed, 4 insertions(+)
create mode 100644 CHANGELOG.md

```

Рис. 4.21: Commit changes

Зальём релизную ветку в основную ветку и отправим данные на GitHub (рис. 4.22).

```

srgubayjdullina@srgubayjdullina:~$ git flow release finish 1.0.0
Branches 'develop' and 'origin/develop' have diverged.
And local branch 'develop' is ahead of 'origin/develop'.
Переключились на ветку «master»
Эта ветка соответствует «origin/master».
Переключились на ветку «develop»
Ваша ветка опережает «origin/develop» на 1 коммит.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
Merge made by the 'ort' strategy.
remote: This repository moved. Please use the new location:
remote:  git@github.com:srgubayjdullina/git-extended.git
To github.com:srgubayjdullina/-git-extended.git
- [deleted]          release1.0.0
Ветка release1.0.0 удалена (была 1a91dc8a).

Summary of actions:
- Release branch 'release1.0.0' has been merged into 'master'
- The release was tagged '1.0.0'
- Release tag '1.0.0' has been back-merged into 'develop'
- Release branch 'release1.0.0' has been locally deleted; it has been remotely d
eleted from 'origin'
- You are now on branch 'develop'

```

Рис. 4.22: Перемещение релизной ветки в основную

Git push –all (рис. 4.23):

```

Терминал
bash-5.2$ git push --all
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (4/4), 372 байта | 372.00 КиБ/с, готово.
Всего 4 (изменений 1), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (1/1), done.
remote: This repository moved. Please use the new location:
remote:  git@github.com:srgubayjdullina/git-extended.git
To github.com:srgubayjdullina/-git-extended.git
  7a9afa9a..d77b7b48  master -> master
* [new branch]      release1.0.0 -> release1.0.0
bash-5.2$ git push --tags
Everything up-to-date
bash-5.2$

```

Рис. 4.23: Отправка данных на GitHub

Создадим релиз на github. Для этого будем использовать утилиты работы с github (рис. 4.24).

```

✓ Logged in as srgubayjdullina
bash-5.2$ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/srgubayjdullina/git-extended/releases/tag/v1.0.0

```

Рис. 4.24: Релиз 1.0.0

6. Создаю ветку для новой функциональности, чтобы новую функциональность разработать (рис. 4.25).

```

bash-5.2$ git flow feature start feature_branch
Переключились на новую ветку «featurefeature_branch»

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

Now, start committing on your feature. When done, use:

    git flow feature finish feature_branch

bash-5.2$ git flow feature finish feature_branch
Переключились на ветку «develop»
Эта ветка соответствует «origin/develop».
Уже актуально.
Ветка featurefeature_branch удалена (была 7a9afa9a).

Summary of actions:
- The feature branch 'featurefeature_branch' was merged into 'develop'
- Feature branch 'featurefeature_branch' has been locally deleted
- You are now on branch 'develop'

```

Рис. 4.25: Создание ветки новой функциональности

Следующим шагом объединяю ветку feature_branch с develop и затем создаю релиз с версией 1.2.3 (рис. 4.26).

```

- Release tag '1.0.0' has been back-merged into 'develop'
- Release branch 'release1.0.0' has been locally deleted; it has been remotely d
eleted from 'origin'
- You are now on branch 'develop'

srgubayjdullina@srgubayjdullina:~$ git flow release finish 1.2.3
Fatal: Branch 'release1.2.3' does not exist and is required.
srgubayjdullina@srgubayjdullina:~$ git flow release start 1.2.3
Branches 'develop' and 'origin/develop' have diverged.
And local branch 'develop' is ahead of 'origin/develop'.
Переключились на новую ветку «release1.2.3»

Summary of actions:
- A new branch 'release1.2.3' was created, based on 'develop'
- You are now on branch 'release1.2.3'

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

    git flow release finish '1.2.3'

```

Рис. 4.26: Создание релиза 1.2.3

Обновляю номер версии в файле package.json, устанавливаю её в 1.2.3 и создаю журнал изменений (рис. 4.27).

```

srgubayjdullina@srgubayjdullina:~$ standard-changelog
✓ output changes to CHANGELOG.md
srgubayjdullina@srgubayjdullina:~$

```

Рис. 4.27: Создание журнала изменений

Далее добавляю журнал изменений в индекс (рис. 4.28).

```
srgubayjdullina@srgubayjdullina:~$ git add CHANGELOG.md
srgubayjdullina@srgubayjdullina:~$ git commit -am 'chore(site): update changelog'
[release1.2.3 a9924243] chore(site): update changelog
1 file changed, 4 insertions(+)
srgubayjdullina@srgubayjdullina:~$
```

Рис. 4.28: Добавление в индекс

Заливаю релизную ветку в основную ветку и отправляю данные на GitHub (рис. 4.29).

```
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (6/6), 761 байт | 761.00 КиБ/с, готово.
Всего 6 (изменений 2), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (2/2), done.
remote: This repository moved. Please use the new location:
remote:   git@github.com:srgubayjdullina/git-extended.git
To github.com:srgubayjdullina/-git-extended.git
   d77b7b48..954a1388  master -> master
srgubayjdullina@srgubayjdullina:~$ git push --all
Всего 0 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: This repository moved. Please use the new location:
remote:   git@github.com:srgubayjdullina/git-extended.git
To github.com:srgubayjdullina/-git-extended.git
   7a9afa9a..1b692ee0  develop -> develop
* [new branch]         release1.2.3 -> release1.2.3
srgubayjdullina@srgubayjdullina:~$
```

Рис. 4.29: Перемещение ветки и отправка на сервер

Git push –tags (рис. 4.30):

```
srgubayjdullina@srgubayjdullina:~$ 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 использовано пакетов 0
remote: This repository moved. Please use the new location:
remote:   git@github.com:srgubayjdullina/git-extended.git
To github.com:srgubayjdullina/-git-extended.git
   * [new tag]         1.0.0 -> 1.0.0
srgubayjdullina@srgubayjdullina:~$
```

Рис. 4.30: Отправка на сервер

Завершаю работу тем, что создаю релиз на github с комментарием из журнала изменений (рис. 4.31).

```
srgubayjdullina@srgubayjdullina:~$ gh release create v1.2.3 -F CHANGELOG.md  
https://github.com/srgubayjdullina/git-extended/releases/tag/v1.2.3  
srgubayjdullina@srgubayjdullina:~$
```

Рис. 4.31: Создание релиза с комментарием

5 Выводы

В ходе выполнения лабораторной работы я получила необходимые навыки правильной работы с репозиториями git.

Список литературы

Лабораторная работа №4: <https://esystem.rudn.ru/mod/page/view.php?id=1098794>