Лабораторная работа №4

Продвинутое использование git

Приспешкин Андрей Андреевич

Содержание

# 1 Цель работы

Получение навыков правильной работы с репозиториями git.

# 2 Задание

Выполнить работу для тестового репозитория.

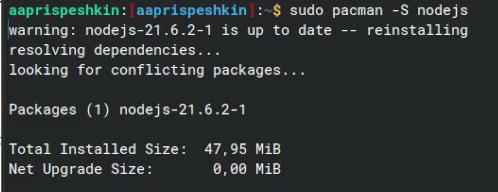
Преобразовать рабочий репозиторий в репозиторий с git-flow и conventional commits.

# 3 Выполнение лабораторной работы

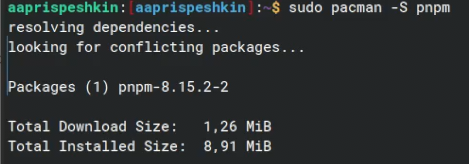
Установим программное обеспечние gitflow node.js и pnpm(Рис.1, 2 и 3)

Установка gitflow пакетным менеджером paru

Установка gitflow пакетным менеджером paru

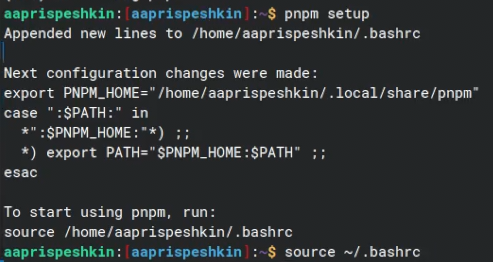


Установка node.js



Установка pnpm

Добавим каталог с исполняемыми файлами через pnpm setup и выполним команду source в каталоге bashrc(Рис.4)



Добавление каталога с исполняемыми файлами

Установим программы commitizen и standard-changelog через pnpm(Рис.5 и 6)

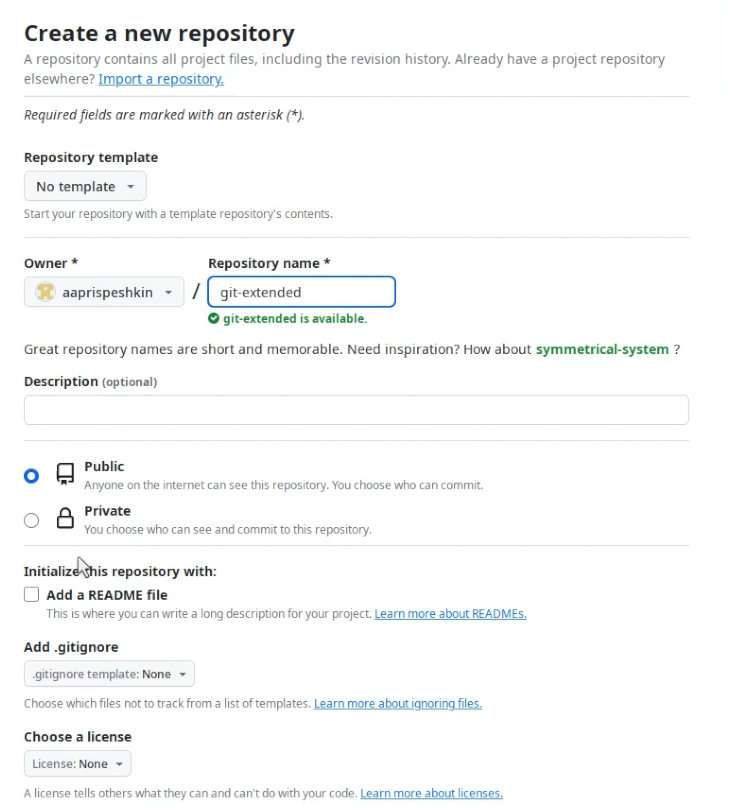
Установка программы commitizen

Установка программы commitizen

Установка программы standard-changelog

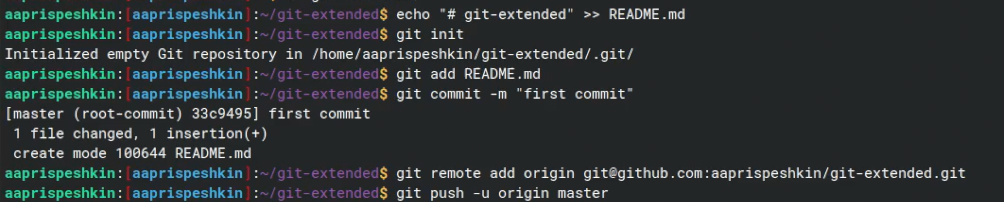
Установка программы standard-changelog

Создадим новый репозиторий на Github и назовём его git-extended(Рис.7)



Создание нового репозитория

Сделаем первый коммит и выложим его на Github(Рис.8)



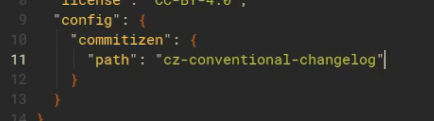
Выкладывание первого коммита на Github

Сконфигурируем пакеты для Node.js через команду pnpm init(Рис.9)

Конфигурация для пакетов Node.js

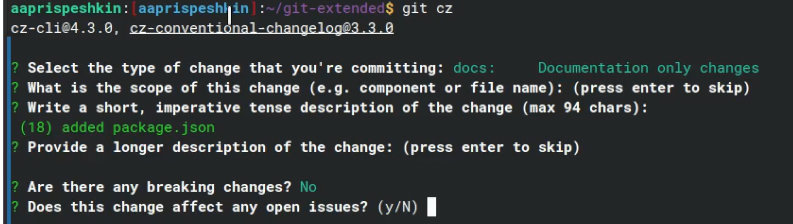
Конфигурация для пакетов Node.js

Добавим в package.json команду для формирования коммитов(Рис.10)



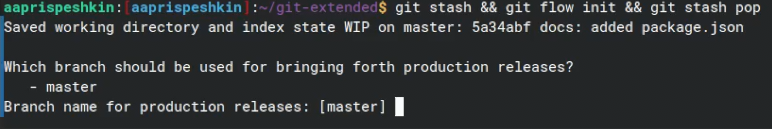
Добавление команды для формирования коммитов

Выполним коммит командой git cz(Рис.11)



Выполнение коммита командой git cz

Начнём конфигурацию gitflow(Рис.12)



Конфигурация gitflow через git flow init

Через git branch проверим что мы находимся на ветке develop(Рис.13)

Проверка текущей ветки с помощью git branch

Проверка текущей ветки с помощью git branch

Установим внешнюю ветку как вышестоящую и создадим релиз с версией 1.0.0(Рис.14)

Установка внешней ветки и создание релиза

Установка внешней ветки и создание релиза

Создадим журнал изменений и добавим его в индекс(Рис.15)



Создание журнала изменений и добавление его в индекс

Зальём релизную ветку в основную ветку через git flow release finish и номер нашего релиза(Рис.16)

Заливаем релизную ветку в основную

Заливаем релизную ветку в основную

Создадим релиз на github утилитой gh(Рис.17)

Создание релиза на github

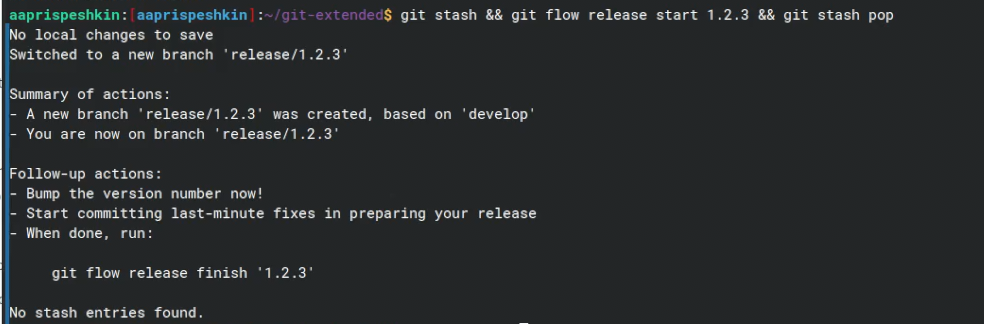
Создание релиза на github

Создадим ветку для новой функциональности и объединим её с веткой develop(Рис.18)



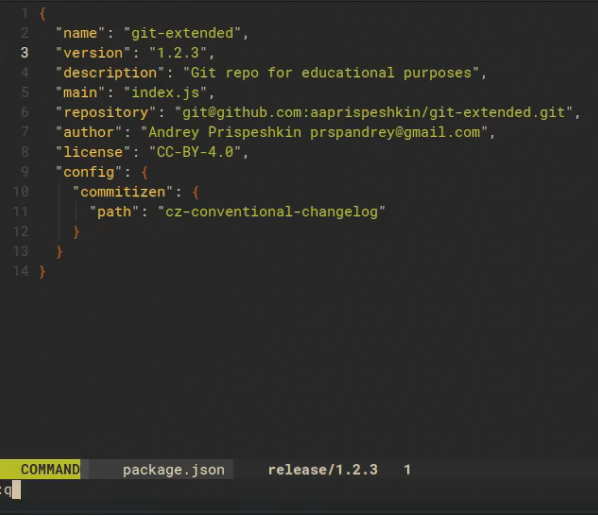
Создание и объединение ветки для новой функциональности

Создадим релиз с версией 1.2.3(Рис.19)



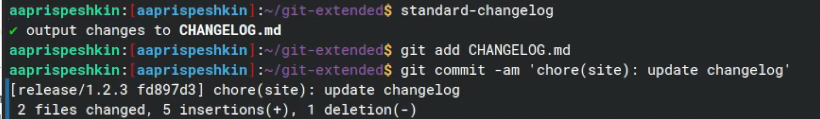
Создание релиза с версией 1.2.3

Обновим номер нашего релиза в package.json(Рис.20)



Обновление номера нашего релиза

Создадим журнал изменений и внесём его в индекс(Рис.21)



Создание журнала изменений и внесение этого журнала в индекс

Зальём релизную ветку в основную ветку(Рис.22)

Объединение релизной ветки и основной

Объединение релизной ветки и основной

Создадим релиз на github с комментарием из журнала изменений(Рис.23)

Создание релиза на github утилитой gh

Создание релиза на github утилитой gh

# 4 Выводы

Я получил навыки правильной работы с репозиториями git.