Отчет для лабораторной работы 4

Продвинутое использование git

Вишняков Родион Сергеевич

Содержание

# Цель работы

Получение навыков правильной работы с репозиториями git.

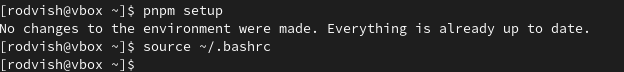
# Теоретические сведения

* Gitflow Workflow опубликована и популяризована Винсентом Дриссеном.
* Gitflow Workflow предполагает выстраивание строгой модели ветвления с учётом выпуска проекта.
* Данная модель отлично подходит для организации рабочего процесса на основе релизов.
* Работа по модели Gitflow включает создание отдельной ветки для исправлений ошибок в рабочей среде.
* Последовательность действий при работе по модели Gitflow:
* Из ветки master создаётся ветка develop.
* Из ветки develop создаётся ветка release.
* Из ветки develop создаются ветки feature.
* Когда работа над веткой feature завершена, она сливается с веткой develop.
* Когда работа над веткой релиза release завершена, она сливается в ветки develop и master.
* Если в master обнаружена проблема, из master создаётся ветка hotfix.
* Когда работа над веткой исправления hotfix завершена, она сливается в ветки develop и master.

# Выполнение лабораторной работы

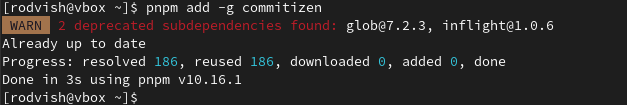
## Работа с тестовым репозиторием

Для работы с Node.js добавим каталог с исполняемыми файлами, устанавливаемыми yarn, в переменную PATH.



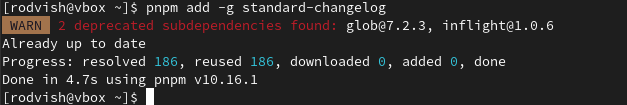
Node.js

Программа commitizen используется для помощи в форматировании коммитов. При этом устанавливается скрипт git-cz, который мы и будем использовать для коммитов.



установка commitizen

Программа standard-changelog используется для помощи в создании логов.

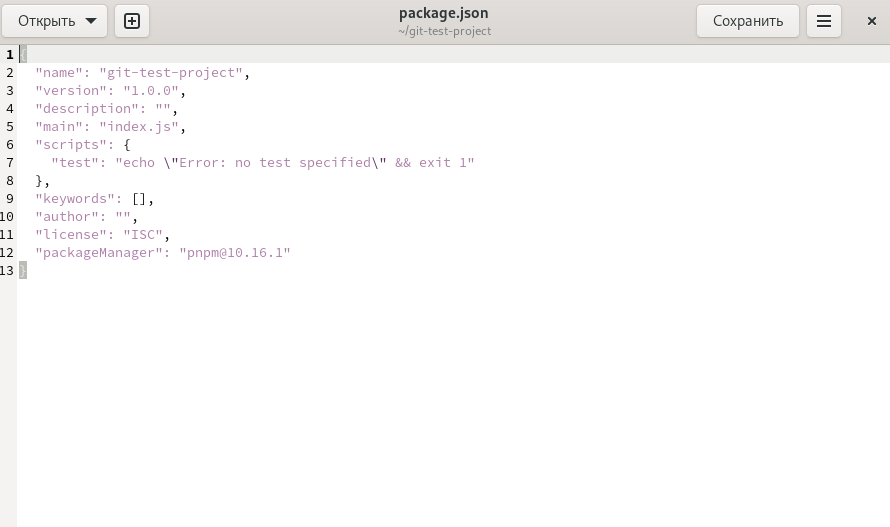


установка standard-changelog

Делаем первый коммит и выкладываем на github.

Необходимо заполнить несколько параметров пакета.

Таким образом, файл package.json приобретает вид:

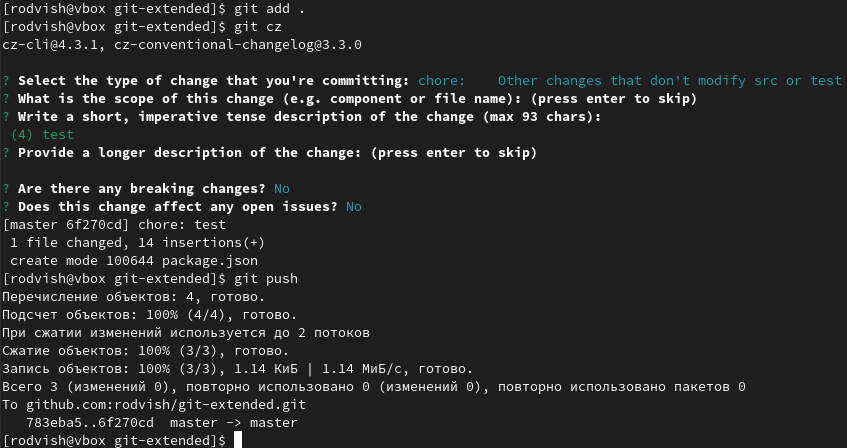


package.json

Добавим новые файлы.

Выполним коммит.

Отправим на github.

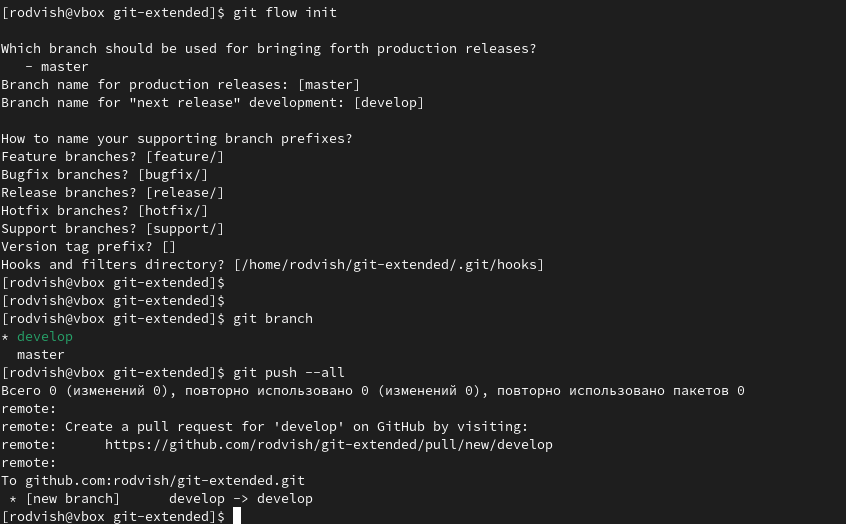


Отправка

Инициализируем git-flow

Проверьте, что Вы на ветке develop

Загрузите весь репозиторий в хранилище



Инициализация и загрузка изменений

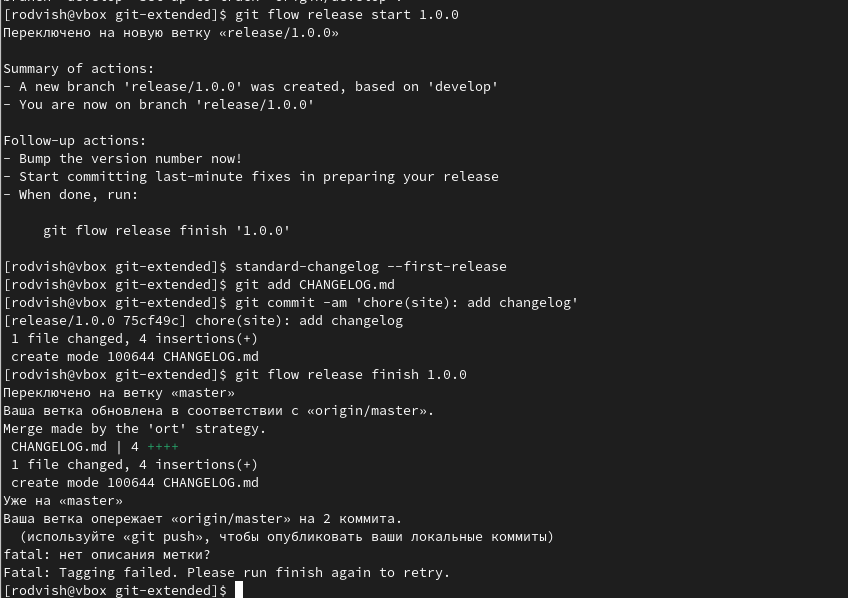
Установите внешнюю ветку как вышестоящую для этой ветки

Создадим релиз с версией 1.0.0

Создадим журнал изменений

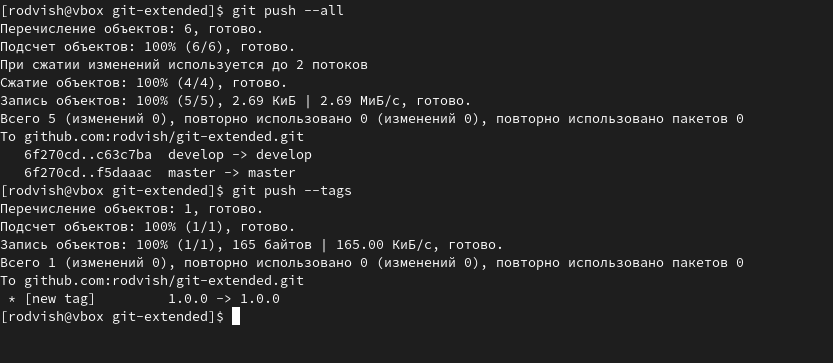
Добавим журнал изменений в индекс

Зальём релизную ветку в основную ветку



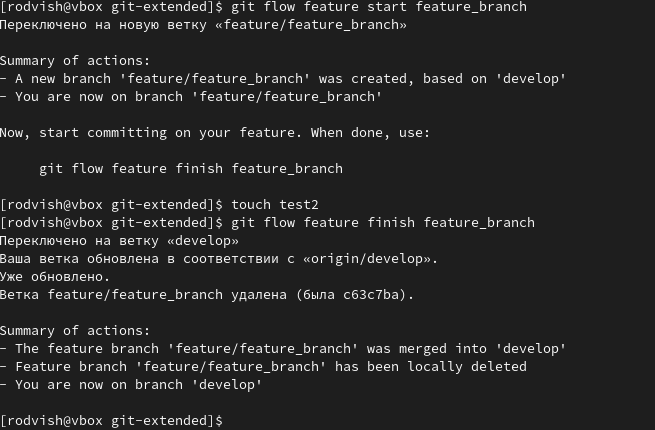
Завершение релиза

Отправим данные на github



Отправка

Создадим ветку для новой функциональности По окончании разработки новой функциональности следующим шагом следует объединить ветку feature\_branch c develop:



Объединение веток

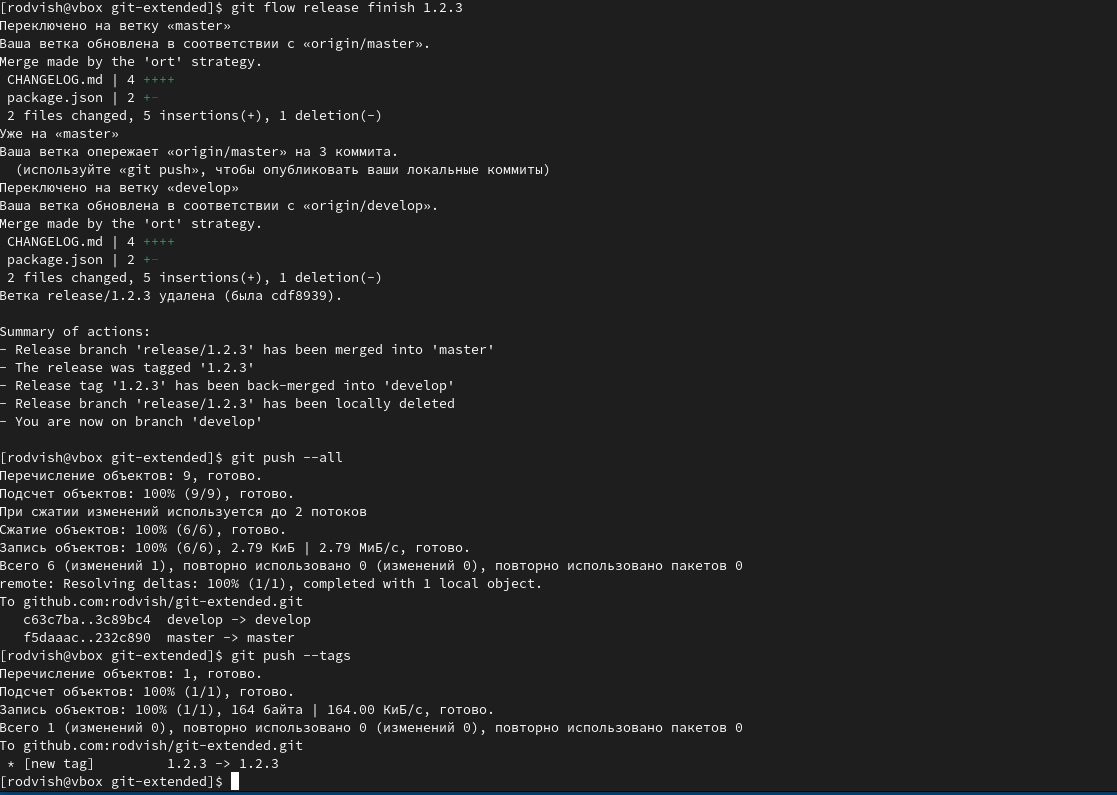
Создадим релиз с версией 1.2.3

Обновите номер версии в файле package.json. Установите её в 1.2.3

Создадим журнал изменений

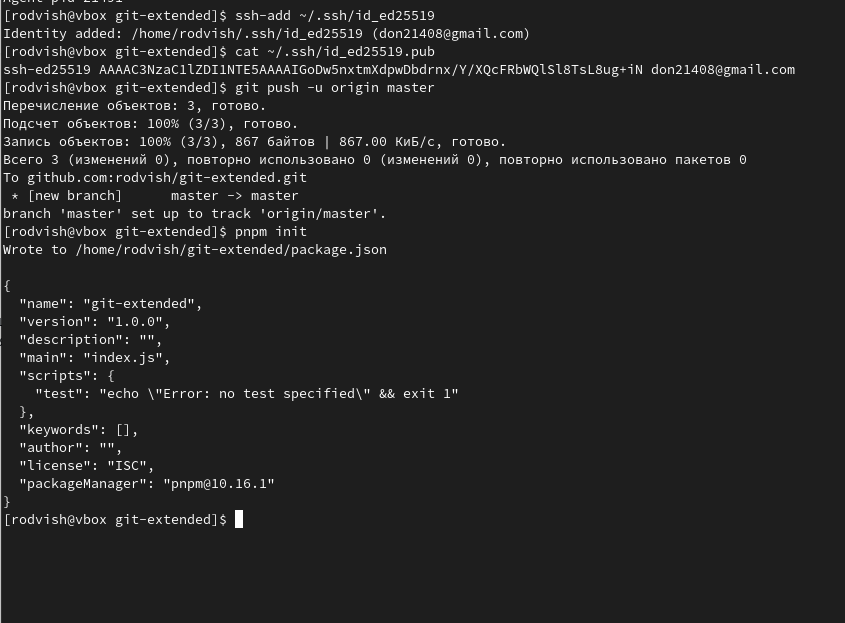
Добавим журнал изменений в индекс

Зальём релизную ветку в основную ветку

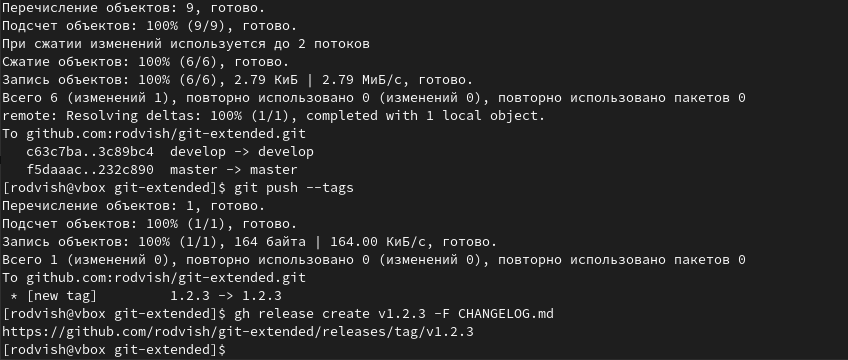


Завершение релиза

## Подготовка рабочего репозитория



package.json и коммит



Завершение релиза

# Вывод

Мы приобрели практические навыки взаимодействия с дополнительными функциями гитхаб.