Лабораторная работа №4

НКАбд-06-23

Улитина Мария Максимовна

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Получение навыков правильной работы с репозиториями git.

2 Задание

1. Выполнить работу для тестового репозитория.
2. Преобразовать рабочий репозиторий в репозиторий с git-flow и conventional commits.

3 Теоретическое введение

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.

4 Выполнение лабораторной работы

Установим gitflow (рис. 4.1).

Рис. 4.1: giflow

(рис. 4.2)

Рис. 4.2: giflow

Настроим node.js

(рис. 4.3)

Рис. 4.3: node.js

(рис. 4.4)

Рис. 4.4: node.js

Установим программу, используемую для помощи в форматировании коммитов (рис. 4.5)

Рис. 4.5: commitizen

Установим программу, используемую для помощи в создании логов (рис. 4.6)

Рис. 4.6: standard changelog

Создадим каталог для репозитория и перейдем в него (рис. 4.7)

Рис. 4.7: каталог

Создадим репозиторий (рис. 4.8)

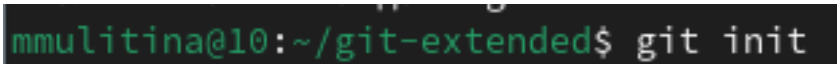

Рис. 4.8: репозиторий

Отправим файл в репозиторий (рис. 4.9)

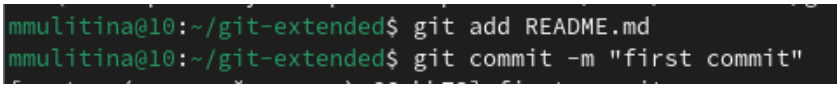

Рис. 4.9: файл

Отправим изменения на сервер (рис. 4.10)

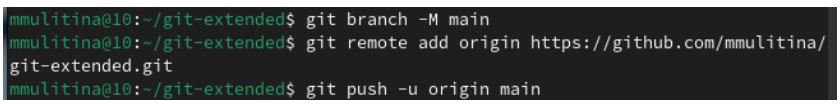

Рис. 4.10: git push

Создадим конфигурацию для пакетов Node.js (рис. 4.11)

Рис. 4.11: Node.js

Изменим конфигурацию и отправим файлы на сервер (рис. 4.12)

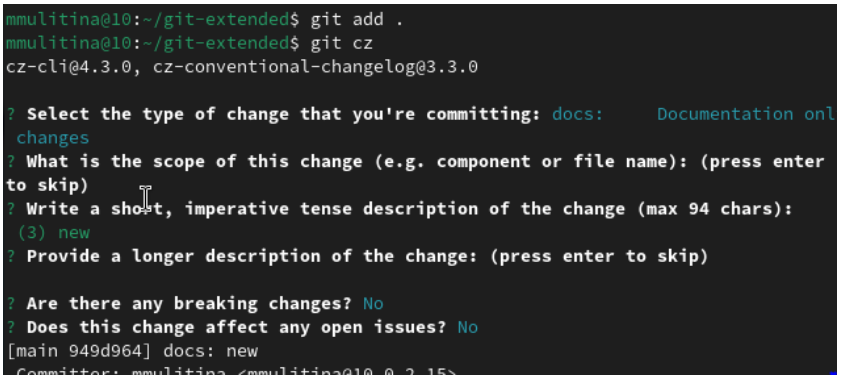

Рис. 4.12: отправка

Инициализируем git-flow (рис. 4.13)

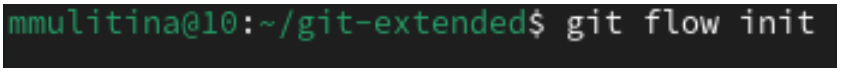

Рис. 4.13: git-flow

Проверим, что мы на ветке develop (рис. 4.14)

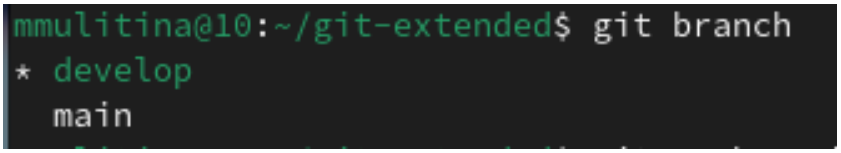

Рис. 4.14: develop

Загрузим весь репозиторий в хранилище (рис. 4.15)

Рис. 4.15: хранилище

Установим внешнюю ветку как вышестоящую для этой ветки (рис. 4.16)

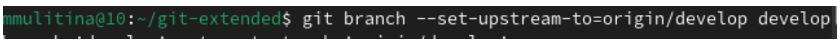

Рис. 4.16: ветка

Создадим релиз с версией 1.0.0 (рис. 4.17)

Рис. 4.17: версия релиза

Создадим журнал изменений и добавим журнал изменений в индекс (рис. 4.18)

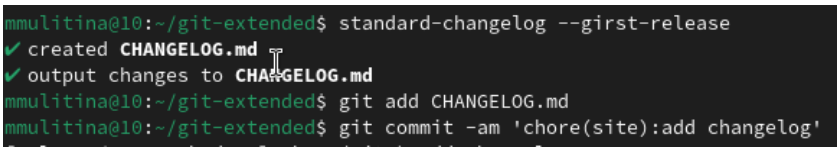

Рис. 4.18: Журнал изменений

Зальём релизную ветку в основную ветку (рис. 4.19)

Рис. 4.19: Основная ветка

Отправим данные на github (рис. 4.20)

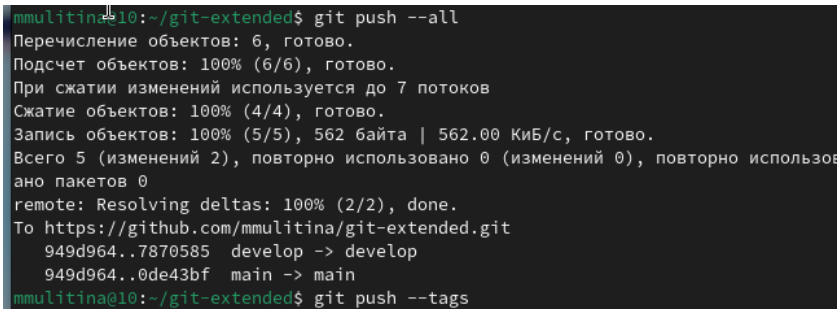

Рис. 4.20: данные на github

Создадим релиз на github. Для этого будем использовать утилиты работы с github (рис. 4.21)

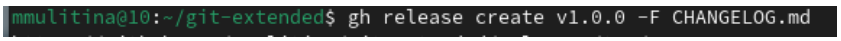

Рис. 4.21: релиз на github

4.1 Работа с репозиторием git

Создадим ветку для новой функциональности (рис. 4.22)

Рис. 4.22: ветка для новой функциональности

По окончании разработки новой функциональности следующим шагом следует объединить ветку feature_branch с develop (рис. 4.23)

Рис. 4.23: объединение

Создадим релиз с версией 1.2.3: (рис. 4.24)

Рис. 4.24: релиз

Создадим журнал изменений (рис. 4.25)

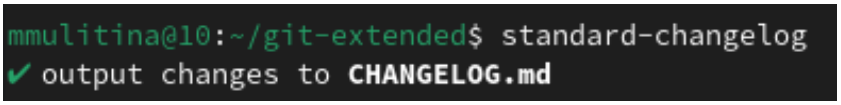

Рис. 4.25: журнал изменений

Добавим журнал изменений в индекс (рис. 4.26)

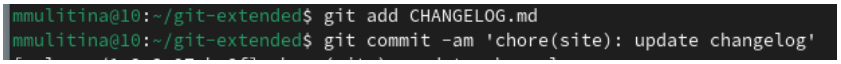

Рис. 4.26: журнал изменений

Зальём релизную ветку в основную ветку (рис. 4.27)

Рис. 4.27: основная ветка

Отправим данные на github (рис. 4.28)

Рис. 4.28: данные на github

(рис. 4.29)

Рис. 4.29: данные на github

(рис. 4.30)

Рис. 4.30: данные на github

Создадим релиз на github с комментарием из журнала изменений (рис. 4.31)

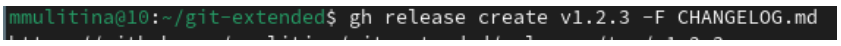

Рис. 4.31: релиз на github

5 Выводы

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

Список литературы

1. Лабораторная работа №4.