Лабораторная работа №4

НКАбд-06-23

Улитина Мария Максимовна

Содержание

# 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. 1).

giflow

Рис. 1: giflow

(рис. 2)

giflow

Рис. 2: giflow

Настроим node.js

(рис. 3)

node.js

Рис. 3: node.js

(рис. 4)

node.js

Рис. 4: node.js

Установим программу, используемую для помощи в форматировании коммитов (рис. 5)

commitizen

Рис. 5: commitizen

Установим программу, используемую для помощи в создании логов (рис. 6)

standard changelog

Рис. 6: standard changelog

Создадим каталог для репозитория и перейдем в него (рис. 7)

каталог

Рис. 7: каталог

Создадим репозиторий (рис. 8)

репозиторий

Рис. 8: репозиторий

Отправим файл в репозиторий (рис. 9)

файл

Рис. 9: файл

Отправим изменения на сервер (рис. 10)

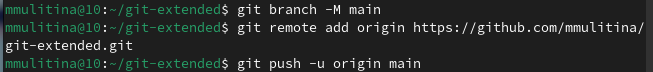


Рис. 10: git push

Создадим конфигурацию для пакетов Node.js (рис. 11)

Node.js

Рис. 11: Node.js

Изменим конфигурацию и отправим файлы на сервер (рис. 12)

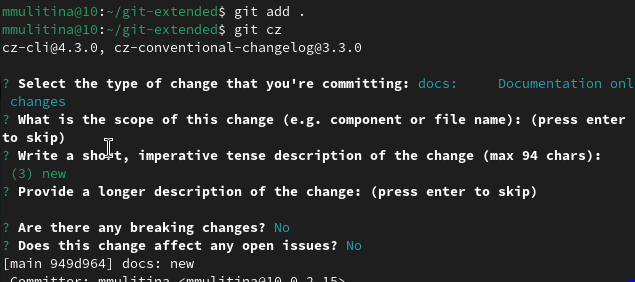


Рис. 12: отправка

Инициализируем git-flow (рис. 13)

git-flow

Рис. 13: git-flow

Проверим, что мы на ветке develop (рис. 14)

develop

Рис. 14: develop

Загрузим весь репозиторий в хранилище (рис. 15)

хранилище

Рис. 15: хранилище

Установим внешнюю ветку как вышестоящую для этой ветки (рис. 16)

ветка

Рис. 16: ветка

Создадим релиз с версией 1.0.0 (рис. 17)

версия релиза

Рис. 17: версия релиза

Создадим журнал изменений и добавим журнал изменений в индекс (рис. 18)

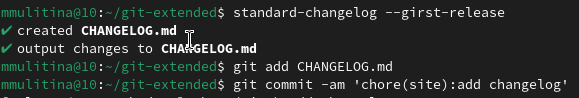


Рис. 18: Журнал изменений

Зальём релизную ветку в основную ветку (рис. 19)

Основная ветка

Рис. 19: Основная ветка

Отправим данные на github (рис. 20)

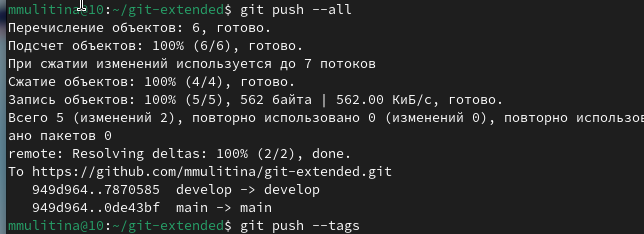


Рис. 20: данные на github

Создадим релиз на github. Для этого будем использовать утилиты работы с github (рис. 21)

релиз на github

Рис. 21: релиз на github

## 4.1 Работа с репозиторием git

Создадим ветку для новой функциональности (рис. 22)

ветка для новой функциональности

Рис. 22: ветка для новой функциональности

По окончании разработки новой функциональности следующим шагом следует объединить ветку feature\_branch c develop (рис. 23)

объединение

Рис. 23: объединение

Создадим релиз с версией 1.2.3: (рис. 24)

релиз

Рис. 24: релиз

Создадим журнал изменений (рис. 25)

журнал изменений

Рис. 25: журнал изменений

Добавим журнал изменений в индекс (рис. 26)

журнал изменений

Рис. 26: журнал изменений

Зальём релизную ветку в основную ветку (рис. 27)

основная ветка

Рис. 27: основная ветка

Отправим данные на github (рис. 28)

данные на github

Рис. 28: данные на github

(рис. 29)

данные на github

Рис. 29: данные на github

(рис. 30)

данные на github

Рис. 30: данные на github

Создадим релиз на github с комментарием из журнала изменений (рис. 31)

релиз на github

Рис. 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.