

Скриншоты репозитория

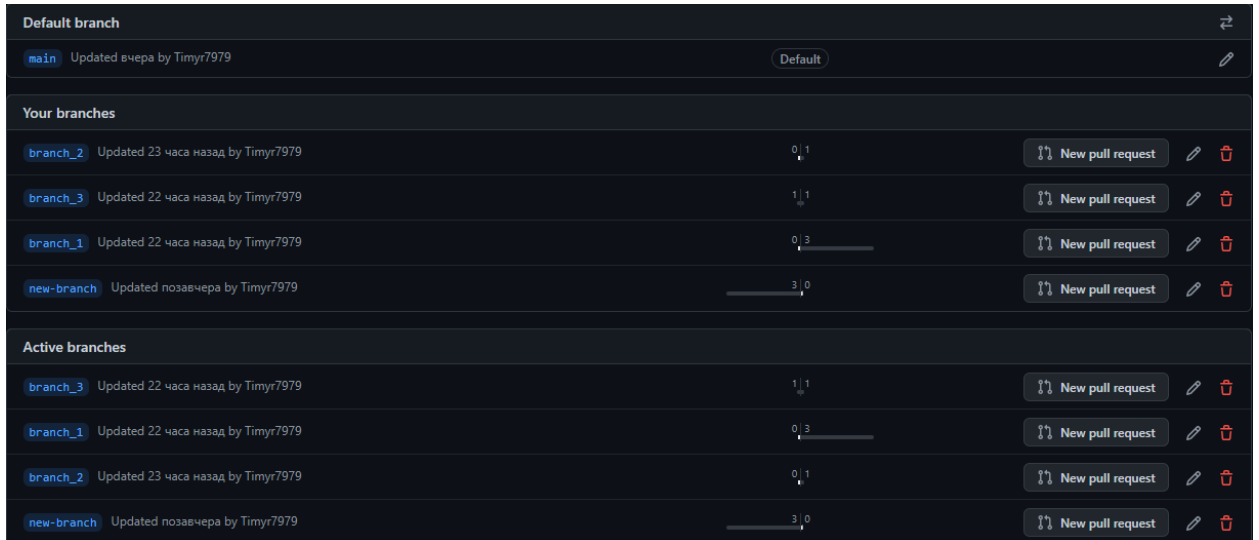


Рисунок 1.1 – ветки

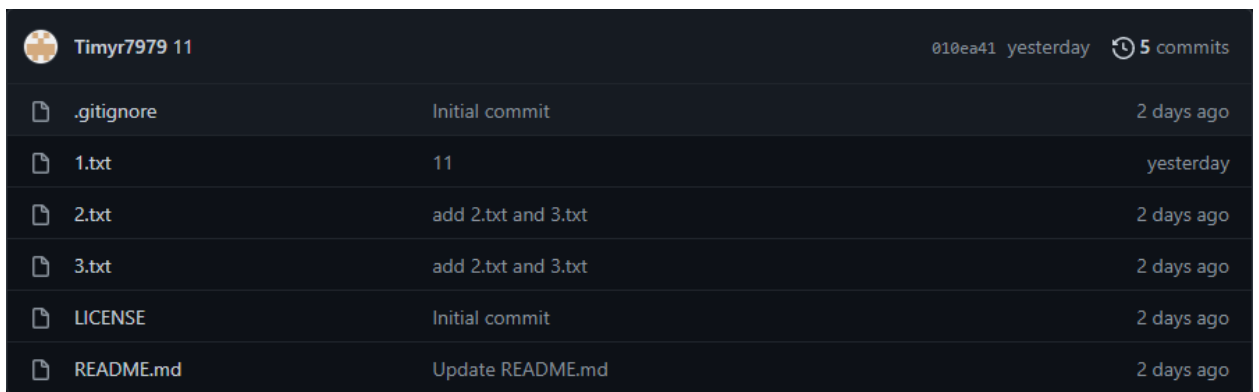


Рисунок 1.1 – файлы в главной ветке

Скриншоты командной строки

```
C:\Users\Тимур\Desktop\gittset\lr3>git checkout branch_2
Switched to branch 'branch_2'
```

Рисунок 2.1 – переключение веток

```
C:\Users\Тимур\Desktop\gittset\lr3>git push origin main
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Timyr7979/lr3.git
5a0cbba..010ea41  main -> main
```

Рисунок 2.2 – отправка изменений ветки на удаленный репозиторий

```
C:\Users\Тимур\Desktop\gittset\lr3>git checkout branch_3
Switched to a new branch 'branch_3'
Branch 'branch_3' set up to track remote branch 'branch_3' from 'origin'.
```

Рисунок 2.3 – создание новой ветки и переключение на нее

ОТВЕТЫ НА КОТРОЛЬНЫЕ ВОПРОСЫ

1. Ветка – это простой перемещаемый указатель на один из коммитов
2. HEAD – это указатель, задача которого ссылаться на определенный коммит в репозитории. Определенный коммит тут, это или родитель следующего коммита, или коммит, относительно которого будет создана рабочая копия во время операции checkout.
3. Ветку можно создать командой `git branch` (создание ветки, но не переключение), `git checkout -b` или через github.
4. Узнать текущую ветку можно переключившись на неё командой `git checkout`.
5. Переключиться на ветку можно командой `git checkout`. Если добавить параметр `-b`, то ветка не только переключится, но и создастся.
6. Удаленная ветка – это ссылка (указатель) в удаленном репозитории, которая включает ветки, теги и т.д
7. Ветки отслеживания – это локальные ветки, которые связаны и удаленной веткой
8. Ветку отслеживания можно создать через команду `–track`
9. Изменения из локальной ветки в удаленную можно отправить по команде `git push`
10. `Git fetch` получает все изменения, которых у вас еще нет, но не будет изменять состояние рабочей директории. Команда `git fetch` – это грубо говоря порядок команд `git fetch` и `git merge`
11. Удалить удаленную ветку можно через команду `git push origin –delete`. Удалить локальную ветку можно через команду `git branch –d`
12. Типы веток `git flow`: функциональная, магистральная, ветка разработки, главная ветка, ветки выпуска, ветки исправления. В `Git-flow` используется больше веток, каждая из которых существует дольше, а коммиты обычно крупнее. Разработчики создают функциональную ветку и откладывают ее слияние с главной магистральной веткой до завершения работы над функцией. `Git-flow` не слишком удобно применять в процессах CI/CD.
13. В SM ветки представлены в виде линий разного цвета. По ним удобно смотреть, когда ветки были созданы, слиты, изменены и т.д. Для переключения веток есть отдельное окно.

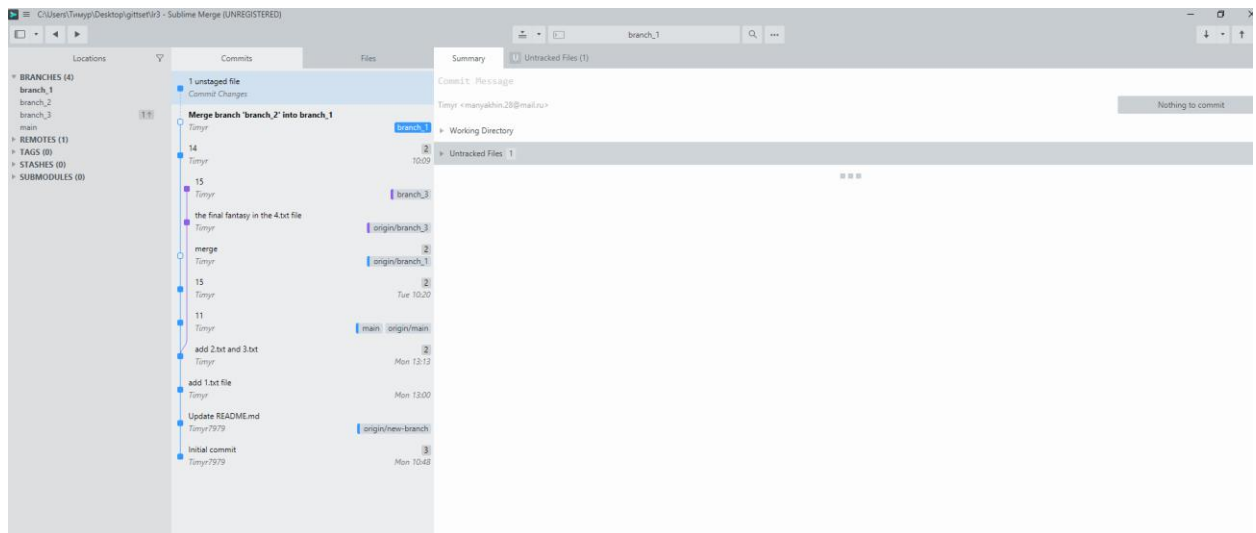


Рисунок 3.1 – рабочее пространство

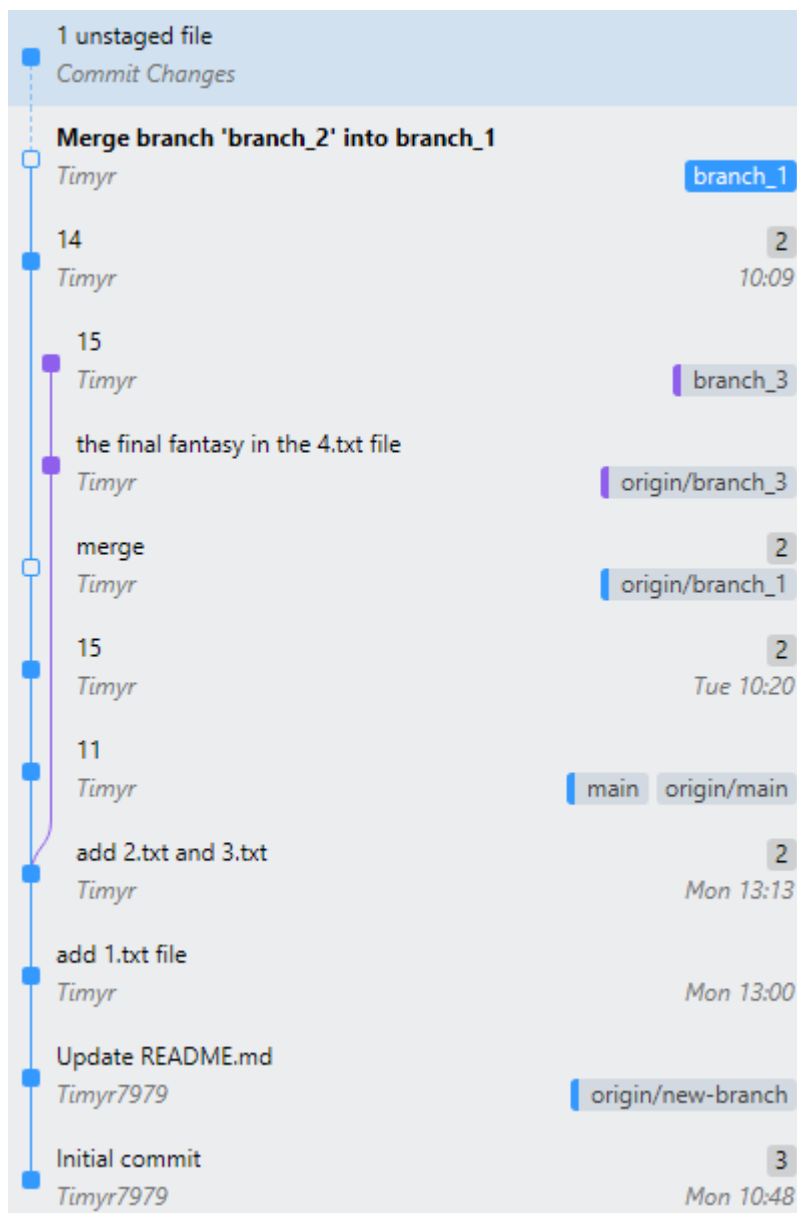


Рисунок 3.2 – ветки в sublime merge