

Отчет по работе с GitHub Desktop

Введение

GitHub Desktop — это кроссплатформенное приложение с графическим интерфейсом для работы с Git и GitHub. Оно предоставляет удобный визуальный интерфейс для выполнения наиболее распространенных операций с репозиториями.

Установка и настройка

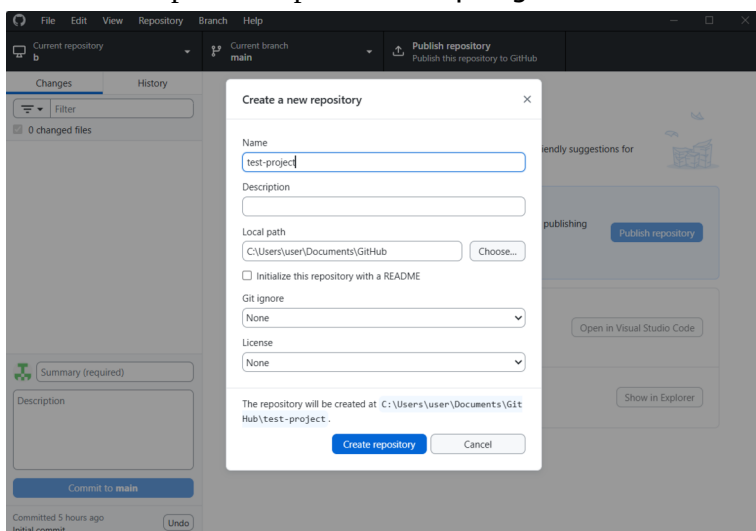
1. Скачал приложение с официального сайта GitHub
2. Выполнил вход в свою учетную запись GitHub
3. Настроил имя и email для коммитов

Основные возможности и команды

1. Создание репозитория

Действие: File → New repository

- Указал имя репозитория: test-project

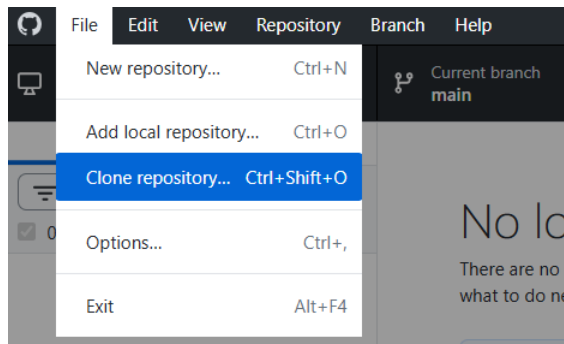


- Выбрал локальный путь
- Можно добавить [README.md](#) и лицензию MIT
- Создал репозиторий кнопкой "Create repository"

Особенность: Автоматически создается начальная структура и выполняется первый коммит.

2. Клонирование репозитория

Действие: File → Clone repository



- Я использовал ссылку на репозиторий, так как не всё вышло настроить, а так же можно выбрать репозиторий из списка своих GitHub-репозиториях.

Clone a repository ×

GitHub.com	GitHub Enterprise	URL
Repository URL or GitHub username and repository (hubot/cool-repo)		
<input type="text" value="URL or username/repository"/>		
Local path		
<input type="text" value="C:\Users\user\Documents\GitHub"/>		<input data-bbox="815 1025 932 1066" type="button" value="Choose..."/>
<div><input data-bbox="523 1133 722 1178" type="button" value="Clone"/> <input data-bbox="734 1133 932 1178" type="button" value="Cancel"/></div>		

- Указал локальный путь для клонирования
- Нажал "Clone"

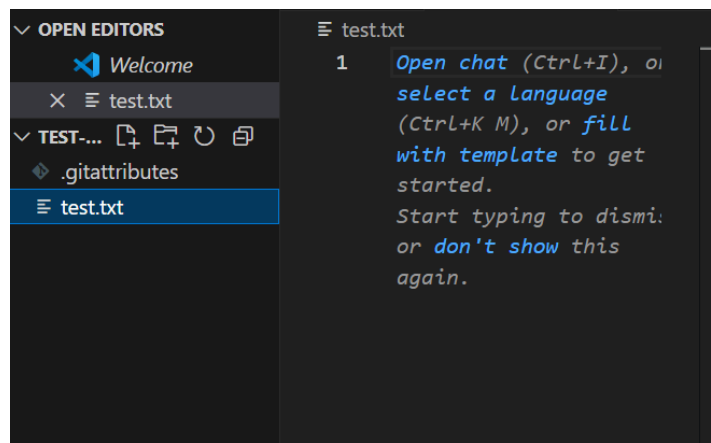
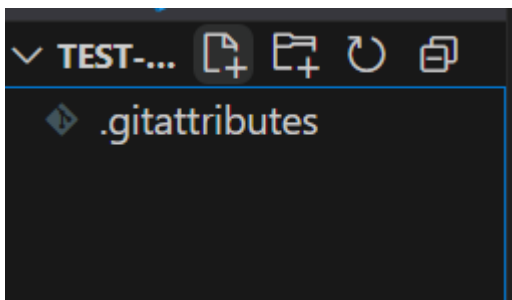
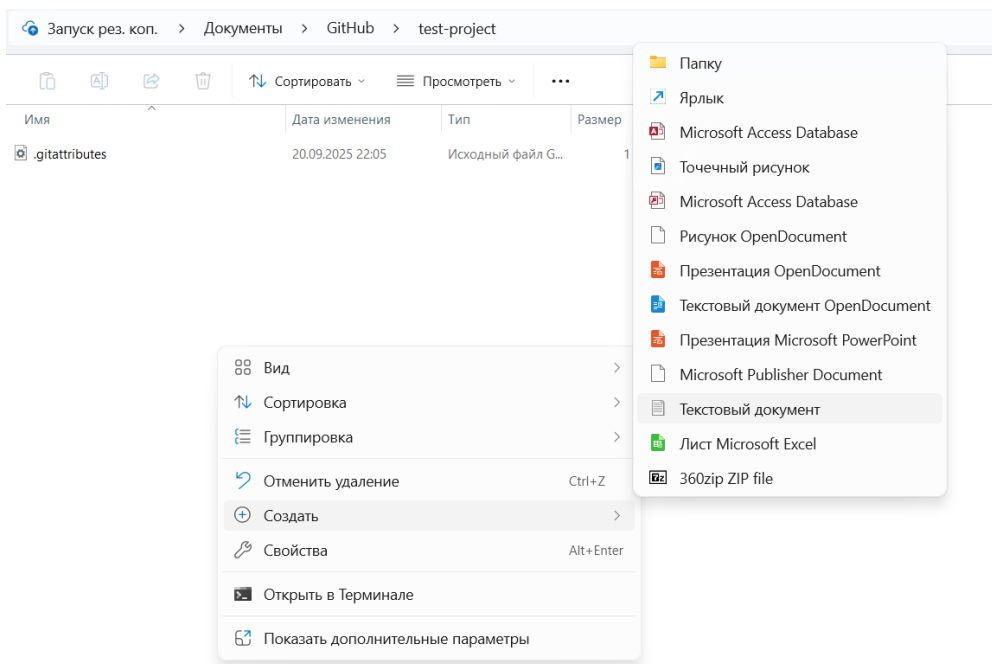
Особенность: Показывает прогресс клонирования и автоматически открывает репозиторий после завершения.

После копирования файлы репозитория можно просмотреть через Проводник или, в моём случае, через Visual Studio Code.

3. Внесение изменений

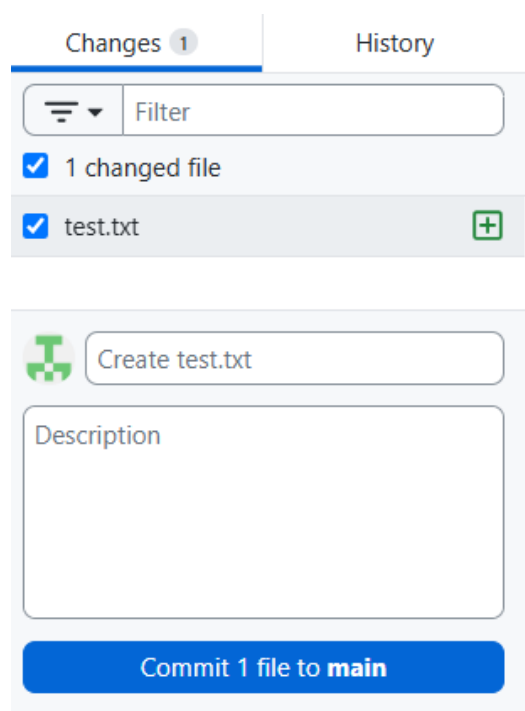
Действие:

- Можно создать файл либо через проводник, либо через текстовый редактор, как VS Code.
- Чтобы создать нужный тебе вид файла, нужно назвать его с соответствующим окончанием, как “.md” для **Markdown** файла, “.txt” для просто текстового файла и тд.
- Сохранить файл



GitHub Desktop автоматически:

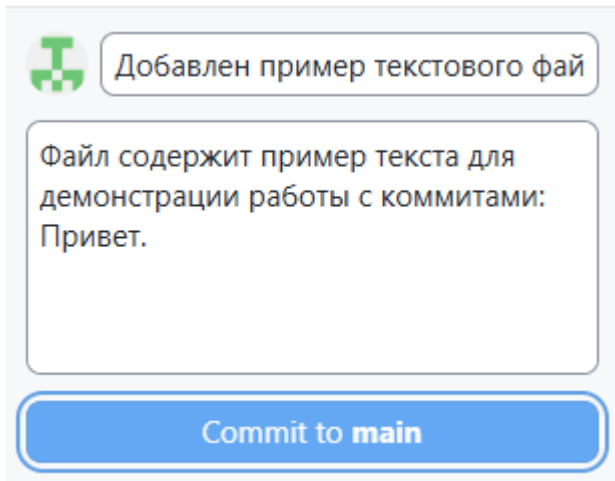
- Обнаружил изменения в разделе "Changes"
- Показал diff изменений
- Предложил создать коммит



4. Создание коммита

Действие:

- В поле "Summary" ввел: "Добавлен пример текстового файла "
- В поле "Description" добавил описание
- Нажал "Commit to main"



Добавлен пример текстового фай

Файл содержит пример текста для демонстрации работы с коммитами: Привет.

Commit to main

Особенность: Визуальное отображение измененных файлов и возможность выборочного добавления изменений в коммит.

Добавлен пример текстового файла

Файл содержит пример текста для демонстрации работы с коммитами: Привет.

Michael322141 ff720f9 +1 -0

1 changed file		test.txt
test.txt		<div>@@ -0,0 +1 @@ 1 + Привет.</div>

5. Работа с ветками

Создание ветки:

- Нажал на выпадающий список текущей ветки
- Выбрал "New branch"
- Ввел имя: feature/new-page

Create a branch

Name

feature/new-page

Your new branch will be based on your currently checked out branch (main). main is the default branch for your repository.

Create branch Cancel

Current branch main

Publish repository Publish this repository

Filter New branch

Default branch

main 20 hours ago

- Нажал "Create branch"

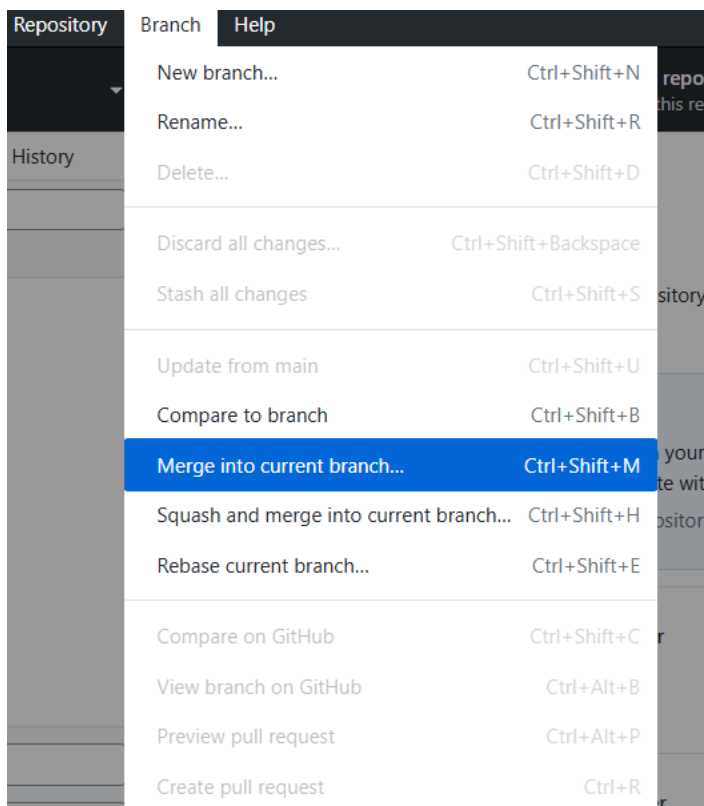
Переключение между ветками:

- Простой выбор из выпадающего списка
- Визуальное отображение текущей ветки

6. Слияние веток

Действие:

- Переключился на ветку `main`
- Нажал "Branch" → "Merge into current branch"



- Выбрал ветку `feature/new-page`
- Нажал "Merge into main"

Особенность: Визуальное отображение процесса слияния и конфликтов, если они возникают.

7. Работа с удаленным репозиторием

Отправка изменений:

- После коммита нажал "Push origin"
- Наблюдал за прогрессом отправки

Получение изменений:

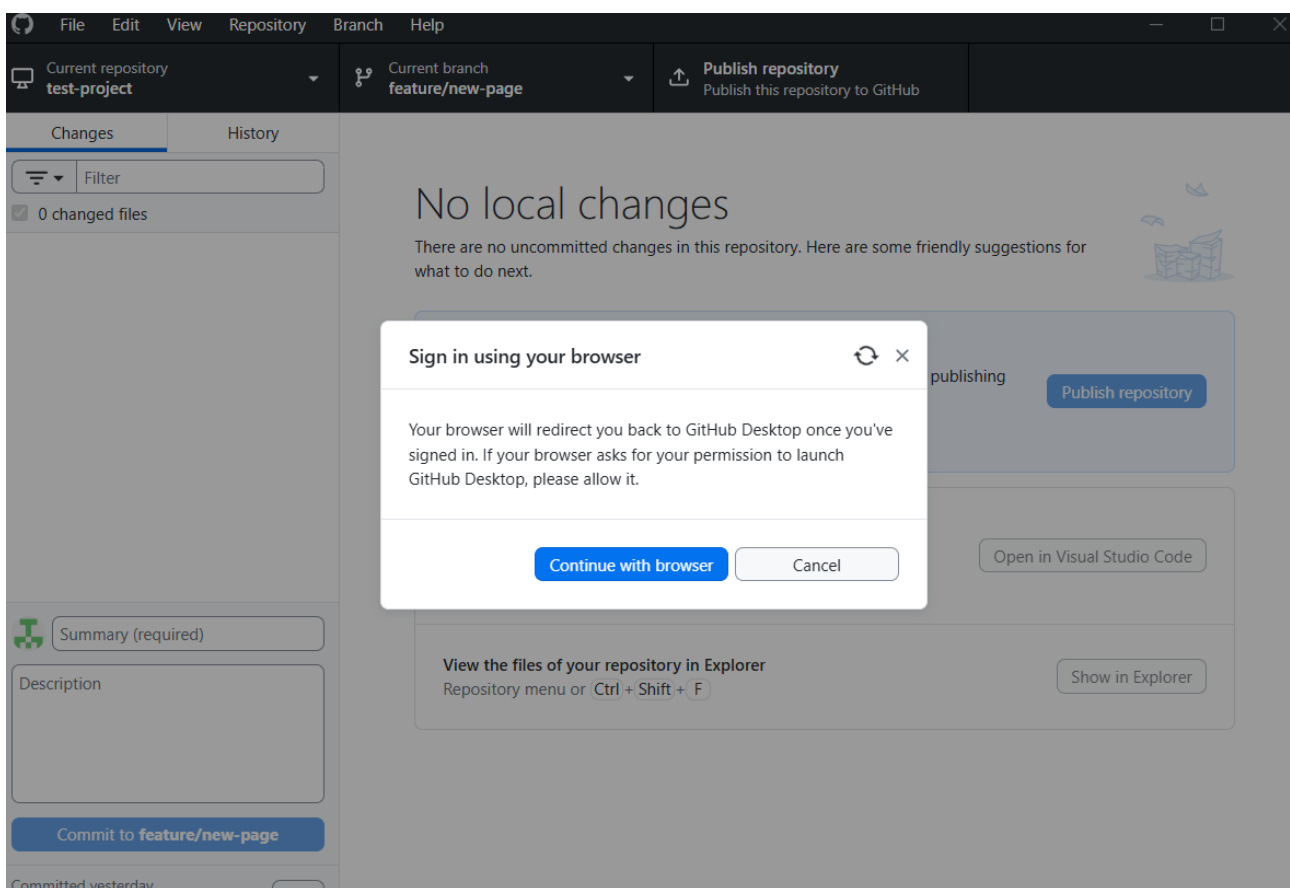
- Нажал "Fetch origin" для проверки обновлений
- При наличии изменений - "Pull origin"

8. Создание Pull Request

Действие:

- После работы в feature-ветке
- Нажал "Create Pull Request"
- Заполнил заголовок и описание PR
- Нажал "Create Pull Request"

Особенность: Должно происходить автоматическое открытие браузера с формой создания PR на GitHub, но у меня из-за технических проблем нет возможности это сделать, так как по какой-то причине приложение не переносит меня в браузер.



9. Разрешение конфликтов

Действие:

- Создать конфликт путем изменения одного файла в двух ветках
- Попытаться выполнить слияние
- GitHub Desktop покажет файлы с конфликтами
- Использовать встроенный редактор для разрешения конфликтов
- Отметить конфликт как разрешенный
- Завершить слияние

Преимущества GitHub Desktop

Для начинающих:

- Интуитивный интерфейс
- Визуальное представление изменений
- Упрощенная работа с ветками
- Автоматическое определение изменений

Для опытных пользователей:

- Быстрое выполнение рутинных операций
- Визуальная история коммитов
- Удобное управление несколькими репозиториями
- Интеграция с GitHub

Особенности:

- Кроссплатформенность (Windows/macOS)
- Встроенный редактор конфликтов
- Подсветка синтаксиса в diff
- Поддержка GitHub Flow
- Интеграция с внешними редакторами кода

Заключение

GitHub Desktop предоставляет отличный визуальный интерфейс для работы с Git, особенно для:

- Начинающих разработчиков
- Команд, использующих GitHub
- Пользователей, которые предпочитают GUI over CLI