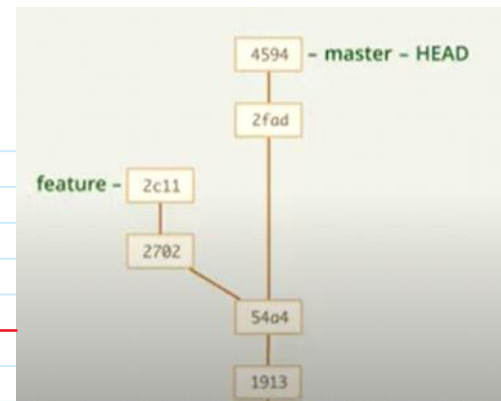


Введение

3 октября 2021 г.

Уровни разработки

- 1) Рабочая область
- 2) Индекс - список измененных файлов для коммита
- 3) Репозиторий - история коммитов. Активная версия



Работа ведется по указателю Head. Он указывает на [ветки](#) или хэш коммита.

Обычно Head можно вызывать через @ , в Винде - в кавычках.

Веток может быть много. Ветки могут сливаться.

Точки на ветках - коммиты. Коммит имеет Хэш и ссылку на предка. Коммит [слияния](#) имеет 2х предков.

Для игнорирования файлов в коммите создают .gitignore

Там записывают маски игнорируемых файлов

/Папка - игнор папки

Раньше основной веткой было - master

Теперь - main

Гит не работает с пустыми папками. Обычно в них кладут .gitkeep

Создание веток основывается на:

Тематические - функционал, сотрудник

Релизные - основной функционал и отшлифованные версии

Git flow - смешанный

Справочник

4 октября 2021 г.

<code>git --help</code>	Все команды (открывает сайт)
<code>git --version</code>	Версия гита

<code>git config --Уровень</code>	Добавление псевдонима команде
<code>alias.Псевдоним 'Команда'</code>	(если в кавычках первым ! то можно связку через ;)

<code>git help Команда</code>	Документация по команде
-------------------------------	-------------------------

Навигация

3 октября 2021 г.

git init Создание репозитория

git config --global user.name "Имя" Добавление глобального автора

git config --global user.name Просмотр имени

git config --global user.email "Почта" Добавление почты

// Уровни

--system Все пользователи системы

--global Текущий пользователь системы

--local Текущий репозиторий

ls -la	Список файлов
-l	- Все папки
-a	- Скрытые файлы
Похоже на <code>dir</code>	

git mv "Старый файл" Перемещение и/или переименование файла
"Новый адрес/ Новое имя"

git rm "Файл" Удаление файла (-f если ругается после изменения)

git rm --cached "Файл" Удаление файла из репозитория, но локально останется

touch "Файл" Создание файла

cd "Каталог" Переход в папку

cd .. Выход на уровень выше

mkdir "Новый каталог" Создание папки

dir Текущая папка

rmdir "Каталог" Удаление пустой папки

git rm -r "Каталог" Удаление папки с файлами

В режиме vim:

1) ESC

:x - Сохранить и выйти

:q - Выйти без сохранения

:w - Сохранить без выхода

vim "Файл"

Изменение файла через Bush

git diff --staged Просмотр изменений из Индекса

git diff Просмотр изменений из рабочей области

git restore "Файл" Откат файла

clear Очистка консоли

git config --list Вывод конфигурации (можно указать явно тип уровня)

cat "Файл" Просмотр файла

Листалка

/слово Поиск строки

n Поиск вперед

Shift+n Поиск назад

q Выход

КОММИТ

4 октября 2021 г. 16:20

<code>git add "Файл1" "Файл2"</code>	Добавление в индекс файлов
<code>git commit -m "Сообщение"</code>	Коммит с сообщением

<code>git commit -am "Сообщение"</code>	Добавление всех файлов и их коммит
-----------------------------------------	------------------------------------

<code>git add .</code>	Добавление всех файлов из текущей директории
<code>git add -A</code>	Добавление всех файлов

<code>git commit -с Коммит</code>	Закоммитит сообщение возьмется из указанного коммита. Откроется редактор
-----------------------------------	--------------------------------------------------------------------------

<code>git commit -C Коммит</code>	То же самое, но без редактора
-----------------------------------	-------------------------------

Коммит можно указать хэшем, веткой, `Orig_head`. Минус: копирует создателя, коммитера и даты

Логи

4 октября 2021 г.

git log	История коммитов
git log -p	С подробностями
git status	Список последних изменений
git show Хэш	Список изменений в коммите

git log --oneline	Короткий вывод списка коммитов
git log "Ветка" --oneline	Вывод списка коммитов в ветке
С флагом -g как git reflog	

git show "Ветка" --quiet	Краткий вывод (без изменений), можно по Хэшу
git show "Ветка"~	Вывод родителя ветки (каждая тильда - шаг назад)
git show "Ветка"~N	На N шагов назад
git show "Ветка": "Файл"	Просмотр файла в ветке
git show : "Файл"	Из Индекса
git show :/ "Описание"	Поиск коммита по слову из сообщения
git reflog "Ветка"	Вывод истории передвижения ветки (кратко, по умолчанию HEAD)

Ветки. Создание

3 октября 2021 г.

Наш указатель всегда HEAD.
Коммиты двигают HEAD, который
двигает за собой указатель ветки

git branch "Новая ветка"	Создание ветки на текущем коммите
git branch -a	Список всех веток
git branch -v	Список веток с указанием хэша коммита
git checkout "Ветка"	Переход на ветку
git checkout Хэш	Переход на коммит (отделенный HEAD - гит удалит позже эту ветку)
git checkout -b "Ветка"	Создание ветки от текущего коммита и переход на нее
git checkout -b "Ветка" Хэш	Создание ветки по хэшу коммита и переход на нее
git checkout -f "Ветка" Хэш	Перенос существующей ветки на коммит (надо уйти с ветки)
git checkout -B "Ветка" Хэш	Перенос существующей ветки на коммит и переход на эту ветку
git checkout -f "Ветка"	Перенос ветки на другой коммит с потерей истории
git checkout "Ветка" "Файл"	Сбросит файл до версии в ветке (можно HEAD)
git checkout Хэш "Файл"	Сбросит файл до версии в коммите
git checkout ""Файл"	Загрузит файл из Индекса
Может быть неопределенность директории, тогда -- перед файлом	
git checkout @{-N}	Перенесет Head на ветку, которая была N чек-аутов назад по рефлогу
git checkout -	Аналогично на 1

Ветки. Удаление

4 октября 2021 г. 16:38

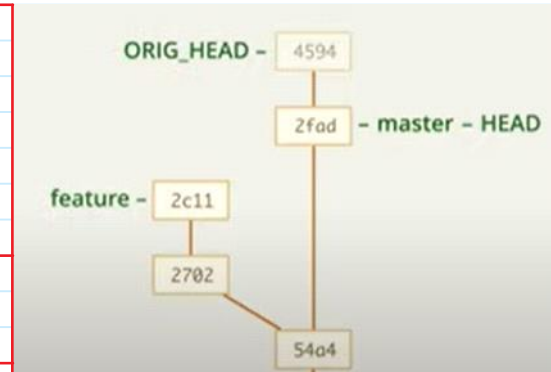
<code>git branch -d "Ветка"</code>	Удаление ветки (имени), коммиты остаются в истории
<code>git branch -D "Ветка"</code>	Удаление ветки (имени), которая не лежит в истории, потеря ветки

<code>git reset --hard Хэш</code>	Перенос HEAD и ветки на другой коммит. Индекс пуст. Файлы вернутся в состояние на коммите, а новые файлы не удалятся. Покидаемый коммит проживет 30 дней. Вместо Хэша можно <code>@~</code> или <code>ORIG_HEAD</code> - предыдущий
-----------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Удаление новых файлов. То, что не делает reset
<code>git clean</code> Флаги
-d - удаление новых папок
-x - удаление новых игнорируемых файлов из <code>.gitignore</code>
-f - обязательный флаг

<code>git reset HEAD "Каталог"</code>	Удаление папки из Индекса
---------------------------------------	---------------------------

<code>git reset --soft Хэш</code>	Перенос HEAD и ветки на другой коммит. Индекс не очистился
-----------------------------------	------------------------------------------------------------



Слияние

3 октября 2021 г.

<code>git remote add "Сервер" "Код репозитория"</code>	Подключение к гиту. Сервер обычно origin
<code>git push "Сервер" "ВеткаСервера"</code>	Отправка изменений на сервер. Или только: <code>git push</code>
<code>git clone "Код репозитория"</code>	Копирование с сервера на локалку в первый раз
<code>git pull "Сервер" "ВеткаСервера"</code>	Получение последнего коммита
<code>git rebase master</code>	
<code>git tag "Тэг"</code>	Создание тега
<code>git tag -a "Тэг" -m "Сообщение"</code>	Создание тега с сообщением
<code>git tag</code>	Показ всех тегов
<code>git push --tags</code>	Отправка всех тегов на сервер
<code>git push "Сервер" "Тэг"</code>	Отправка конкретного тега
<code>git remote -v</code>	Список связанных серверов
<code>git merge "Ветка"</code>	Слияние текущей ветки и указанной
<code>git stash</code>	Архивирование изменений для переключения веток
<code>git stash pop</code>	Загрузка изменений обратно (не привязано к ветке где создано)