

# Работа с изменениями

## Просмотр изменений

### Сводная информация

```
git status
```

### Изменения относительно индекса

```
git diff
```

### Изменения подготовленные для последующей их записи

```
git diff --cached
```

# Смена ветки при наличии изменений

Смена ветки при наличии изменения происходит таким же способом как и обычная смена ветки.

```
git checkout -b <имя_ветки>
```

## Отмена изменений

### Полный сброс

```
git reset --hard <hash_commit or HEAD>
```

### Сброс с сохранением изменений

```
git reset <hash_commit or HEAD>
```

## Отмена изменений заданного файла

```
git checkout <имя_файла>
```

или

```
git checkout <hash_commit or HEAD> -- <имя_файла>
```

# Отправка изменений

```
git push origin <имя_ветки>
```

В случае, когда требуется принудительно отправить изменения в ветку, независимо от её состояния, в конце добавляется параметр `--force`

```
git push origin <имя_ветки> --force
```

# Скрытие изменений

Вы выполнили какие-нибудь изменения в файлах и хотите переключиться на другую ветку, но чтобы там не было ваших текущих изменений. С помощью команды `git stash` можно скрыть эти изменения. Ваши изменения помещаются в отдельное хранилище — в стек, а вы можете спокойно переключиться на другую ветку с дальнейшим извлечением скрытых изменений.

## Скрытие изменений с добавлением комментария

```
git stash save "Комментарий"
```

## Вывести список скрытых изменений

Самые старые скрытые изменения отображаются внизу списка, самые свежие сверху. Каждое скрытое изменение имеет идентификатор с номером, например, `stash@{0}`

```
git stash list
```

## Применение скрытых изменений

Команда `git stash apply` берет самое свежее скрытое изменение (`stash@{0}`) и применяет его к текущему репозиторию. Это похоже на то, как вы применяете патч, только в качестве патча выступает ваше скрытое изменение.

```
git stash apply
```

Можно указать идентификатор для его конкретного применения.

```
git stash apply stash@{<stash_number>}
```

## Применение скрытых изменений с их удалением из скрытого

Команда `git stash pop` сделать всё тоже самое, что и команда `git stash apply`, при этом удалив скрытые изменения из списка скрытых изменений.

```
git stash pop
```

```
git stash pop stash@{<stash_number>}
```

## Обзор скрытых изменений

```
git stash show
```

```
git stash show stash@{<номер_скрытого_изменения>}
```

## Полный список скрытых изменений

```
git stash show -p
```

## Создание отдельной ветки из скрытых изменений

```
git stash branch <новое_имя_ветки>
```

```
git stash branch <новое_имя_ветки> stash@{<номер_скрытого_изменения>}
```

При этом скрытое изменение удаляется из списка скрытых изменений.

## Удаление скрытых изменений

```
git stash drop
```

```
git stash drop stash@{<номер_скрытого_изменения>}
```

## Удаление всех скрытых изменений

```
git stash clear
```

# Список используемых ИСТОЧНИКОВ

- [Полезные команды Git: безопасная отмена коммитов, добавление файла из другой ветки и другие](#)

---

Revision #6

Created 25 June 2023 11:59:50 by Антон Сергеевич Абраменко

Updated 15 February 2024 04:19:33 by Антон Сергеевич Абраменко