

Работа с файлами

№ урока: 3 **Курс:** Основы использования Git

Средства обучения: Git console.

Обзор, цель и назначение урока

Узнать, какие состояния может иметь файл в репозитории Git. Научиться добавлять, изменять, удалять, перемещать и переименовывать файлы. Научиться выключать отслеживание для файлов и директорий.

Изучив материал данного занятия, учащийся сможет:

- Производить любые манипуляции с файлами в репозитории.
- Смотреть изменения файлов в индексе.
- Добавлять файлы в .gitignore.

Содержание урока

1. Состояния файлов
2. Операции с файлами
3. Отключение контроля состояния для файлов и директорий (.gitignore)

Резюме

- Файлы в репозитории делятся на **отслеживаемые** (tracked) и **неотслеживаемые** (untracked).
- Untracked – это новый файл, который не был добавлен для отслеживания командой git add. Все остальные файлы имеют состояние tracked.
- В свою очередь отслеживаемые файлы могут иметь 3 разных состояния: **неизменённое** (unmodified), **изменённое** (modified) и **проиндексированное** (staged).
- Файл находится в неизменённом состоянии если физически файл в директории соответствует сохранённой копии в хранилище git.
- Файл переходит в изменённое когда он под контролем гит и он как-то модифицировался. Важный момент в том что если выполнить коммит - то файлы в изменённом состоянии в него не войдут и не будут зафиксированы.
- Команда git add переводит файлы из состояний неотслеживаемого и изменённого в состояние проиндексирован и только после этой операции изменения сохранятся с выполнением коммита. Это промежуточное состояние нужно потому что не всегда изменённые файлы должны входить в коммит.
- После коммита гит статус снова будет иметь вывод как в первом пункте на слайде.
- **git diff --staged (--cached)** – вывод в консоль изменений, которые сейчас находятся в области индекса.
- **git commit -a -m "[message]"** – команда объединяет два действия – добавление всех изменений в индекс и выполнение коммита (флаг **-a** показывает что все файлы в изменённом состоянии нужно проиндексировать перед коммитом).
- **git rm <file>** - удаляет файл из файловой системы и добавляет это изменение в индекс. **--cached** параметр позволяет удалить файл из кеша гит, но при этом он останется доступным в рабочей директории.
- **git mv <file>** - меняет имя файла либо перемещает файл и добавляет это изменение в индекс.

- **.gitignore** - файл, позволяющий указывать что в репозитории не следует отслеживать. Работает с файлами которые находятся в состоянии неотслеживаемых изначально. Но если нужно отключить контроль для файла, который находился под контролем git то сделать это немного сложнее, нужно использовать команду `git rm --cached <file>`. Или же можно вручную переместить файл, удалить его в git сделав коммит, добавить в .gitignore и вернуть на место.
- Синтаксис .gitignore файла: каждая строка - отдельный шаблон, комментарии должны начинаться с символа "#". Символ "/" в начале строки указывает на текущую папку. Символ "*" заменяет любое количество символов. Символ "!" в начале строки инвертирует шаблон. Пример:

```
# Игнорировать файл foo.txt.
foo.txt
# Игнорировать html файлы
*.html
# Но конкретно foo.html не игнорировать
!foo.html
# Игнорировать rar файлы в корне проекта
# Допустим файл /temp/main.rar не будет проигнорирован т.к. он не в корне
/*.*rar
# Игнорировать css файлы из папки bar не включая подпапки
# Допустим файл /bar/temp/main.css не будет проигнорирован т.к. он в подпапке temp
/bar/*.css
# Игнорировать js файлы из папки bar и подпапок, если таковые будут
/bar/**/*.js
```

Закрепление материала

- Как можно удалить файл в репозитории не используя команду `git rm <file>`?
- Какой командой можно убрать файл из-под контроля Git?
- Какие файлы не следует добавлять в .gitignore?
- В какое состояние перейдет файл после того, как был выполнен его коммит?
- Какую команду можно использовать вместо того, чтобы перемещать или менять имя файла вручную?

Дополнительное задание

Задание

Добавьте файл в репозиторий, внесите изменения, затем добавьте их в индекс. Снова измените файл и выведите `git status`. Объясните результат.

Самостоятельная деятельность учащегося

Задание 1

Создайте новый репозиторий, добавьте несколько файлов так, чтобы они были в состоянии `unmodified`. Переведите их в состояние `staged` и потом выполните коммит. На каждом этапе следите за статусом файлов с командой `git status`.

Задание 2

Удалите вручную файл из репозитория и зафиксируйте это изменение выполнив коммит. Затем выполните удаление командой `git rm`

Задание 3

Поменяйте имя одного из файлов и сохраните это изменение. Затем сделайте то же самое с помощью команды `git mv`

Задание 4

Добавьте в репозиторий файл и папку с файлом внутри, оставьте их в состоянии `untracked`, затем добавьте файл `.gitignore` и включите туда такие условия, чтобы новый файл и папка игнорировались.

Рекомендуемые ресурсы

Официальная документация Git

<https://git-scm.com/docs>

Книга Pro Git, Scott Chacon, Ben Straub

<https://git-scm.com/book/ru/v2>