

BACKEND - РАЗРАБОТКА НА ФРЕЙМВОРКЕ JAVA SPRING



РЭУ ИМ. Г.В. ПЛЕХАНОВА МПТ

www.ucmpt.ru
www.do.rea.ru

СИСТЕМА КОНТРОЛЯ ВЕРСИЯМИ GIT



Git — это система контроля версий, которая помогает отслеживать историю изменений в файлах. Git используют программисты для совместной работы над проектами. Git — система контроля версий. В самом простом виде контроль версий — это сохранение на компьютере серии измененных файлов, например с разными датами в названии, или режим отслеживания исправлений в текстовых документах.



РЕПОЗИТОРИЙ GIT



Репозиторий Git или репозиторий – это папка, в которую Git отслеживает изменения. На компьютере может быть любое количество репозиториев, каждое из которых хранится в собственной папке. Каждый репозиторий Git в системе является независимым, поэтому изменения, сохраненные в одном репозитории Git, не влияют на содержимое другого.



ПЕРВОНАЧАЛЬНЫЕ НАСТРОЙКИ



- После установки git следует его настроить.
- `git config --global user.email "(Ваша почта)"`
- `git config --global user.name "(Ваше имя)"`
- В ином случае, при сохранении состояния git будет требовать вписать почту и имя. Это необходимо в качестве информации, кто данный коммит сделал и как с ним связаться.

СОЗДАНИЕ РЕПОЗИТОРИЯ



- Для инициализации репозитория по пути в котором находимся прописываем следующую команду:
- `git init`



РАБОТА С РЕПОЗИТОРИЕМ

- Для подготовки файлов необходимо прописать команду
- `git add .`
- Для проверки состояния имеется команда
- `git status`
- Для того чтобы сохранить состояние нашего проекта используется команда
- `git commit -m "<Название коммита>"`
- Ключ `-m(message)` нужен для того прописать название коммита, в котором пишется основные изменения.



ВЕТКИ

- Для отображения существующих веток
 - `git branch`
- Для создания новой ветки
 - `git branch <Название ветки>`
- Для переключения на ветку используется `checkout`
 - `git checkout <Название ветки>`

ВЕТКИ



- Также можно одновременно создать ветку и на неё переключится
- `git checkout -b <Название ветки>`
- Для того чтобы объединить ветки
- `git merge <Название ветки> / git merge <Название ветки> -m <сообщение>`
- Следует помнить, что слияние идет из указанного в команде ветки в ветку в которой находитесь.
- Если нужно выйти из состояния слияния
- `git merge --abort`

СЕРВИСЫ GIT



- **GitHub**
- **GitLab**
- **Bitbucket**
- **Gogs**

GIT INIT



git init – инициализация локального репозитория

```
ProjectUsers 1                                Проект
[ilabukanov@MacBook-Pro Курс % git init
подсказка: Using 'master' as the name for the initial branch. This default branch name
подсказка: is subject to change. To configure the initial branch name to use in all
подсказка: of your new repositories, which will suppress this warning, call:
подсказка:
подсказка:     git config --global init.defaultBranch <name>
подсказка:
подсказка: Names commonly chosen instead of 'master' are 'main', 'trunk' and
подсказка: 'development'. The just-created branch can be renamed via this command:
подсказка:
подсказка:     git branch -m <name>
Инициализирован пустой репозиторий Git в /Users/ilabukanov/Documents/JavaProject/Курс/.git/
ilabukanov@MacBook-Pro Курс %
```

GIT STATUS



git status – состояние проекта, измененные и не добавленные файлы, индексированные файлы.

```
Инициализирован пустой репозиторий Git в /Users/ilabukanov/Documents/JavaProject/Курс/.git/
ilabukanov@MacBook-Pro Курс % git status
Текущая ветка: master

Еще нет коммитов

Неотслеживаемые файлы:
(используйте «git add <файл>...», чтобы добавить в то, что будет включено в коммит)
.DS_Store
API.zip
API/
ProjectUsers 1/
SpringSecurityApplication.zip
SpringSecurityApplication/
"\320\237\321\200\320\276\320\265\320\272\321\202 \321\201 \320\277\320\276\320\273\321\214\320\267\320\276\320\262\320\260\321\202\320\265\320\273\321\217\320\274\320\270 \321\201 \320\267\320\260\320\263\321\200\321\203\320\267\320\272\320\276\320\271 \321\204\320\276\321\202\320\276.zip"

индекс пуст, но есть неотслеживаемые файлы
(используйте «git add», чтобы проиндексировать их)
ilabukanov@MacBook-Pro Курс % █
```

GIT ADD



git add – добавление файлов в локальный репозиторий

git rm --cached <файл> удаление файла из репозитория

```
fatal: спецификатор пути «file» не соответствует
[ilabukanov@MacBook-Pro Kypc % git add file.txt
[ilabukanov@MacBook-Pro Kypc % █
```

```
[ilabukanov@MacBook-Pro Kypc % git add file.txt
[ilabukanov@MacBook-Pro Kypc % git status
Текущая ветка: master

Еще нет коммитов

Изменения, которые будут включены в коммит:
(используйте «git rm --cached <файл>...», чтобы убрать из индекса)
    новый файл:   file.txt
```

GIT COMMIT



git commit - это команда для записи индексированных изменений в репозиторий Git

```
ilabukanov@MacBook-Pro Курс % git commit -m "Добавлен новый текстовый файл"
[master (корневой коммит) 69d1f73] Добавлен новый текстовый файл
  Committer: Илья Буканов <ilabukanov@MacBook-Pro.local>
  Ваше имя или электронная почта настроены автоматически на основании вашего
  имени пользователя и имени машины. Пожалуйста, проверьте, что они
  определены правильно.
  Вы можете отключить это уведомление установив их напрямую. Запустите следующую
  команду и следуйте инструкциям вашего текстового редактора, для
  редактирования вашего файла конфигурации:

    git config --global --edit

После этого, изменить авторство этой коммита можно будет с помощью команды:

    git commit --amend --reset-author

1 file changed, 1 insertion(+)
create mode 100644 file.txt
ilabukanov@MacBook-Pro Курс %
```

GIT LOG



git log используется для просмотра истории коммитов, начиная с самого свежего и уходя к истокам проекта

```
(используйте «git add», чтобы проиндексировать их)
[ilabukanov@MacBook-Pro Курс % git log
commit 69d1f73195aa99e9ae23576d509dcb9614b622b8 (HEAD -> master)
Author: Илья Буканов <ilabukanov@MacBook-Pro.local>
Date: Thu Nov 17 07:42:08 2022 +0300

    Добавлен новый текстовый файл
```

```
fatal: неопознанный аргумент: --пкфзр
[ilabukanov@MacBook-Pro Курс % git log --graph
* commit aad715c450278422bd195d293e09d4ffad502d9e (HEAD -> master)
| Author: Илья Буканов <ilabukanov@MacBook-Pro.local>
| Date: Thu Nov 17 07:48:33 2022 +0300
|
|     В текстовый файл добавлено сообщение Вносим новый текст в файле
|
* commit 69d1f73195aa99e9ae23576d509dcb9614b622b8
  Author: Илья Буканов <ilabukanov@MacBook-Pro.local>
  Date: Thu Nov 17 07:42:08 2022 +0300

      Добавлен новый текстовый файл
ilabukanov@MacBook-Pro Курс %
```

GIT CHECKOUT



В Git под термином checkout подразумевают переключение между различными версиями целевого объекта

```
[ilabukanov@MacBook-Pro Kypc % git checkout 69d1f73195aa99e9ae23576d509dcb9614b622b8
Примечание: переключение на «69d1f73195aa99e9ae23576d509dcb9614b622b8».
```

Вы сейчас в состоянии «отсоединённого указателя HEAD». Можете осмотреться, внести экспериментальные изменения и зафиксировать их, также можете отменить любые коммиты, созданные в этом состоянии, не затрагивая другие ветки, переключившись обратно на любую ветку.

Если хотите создать новую ветку для сохранения созданных коммитов, можете сделать это (сейчас или позже), используя команду switch с параметром -с. Например:

```
git switch -с <новая-ветка>
```

Или отмените эту операцию с помощью:

```
git switch -
```

Отключите этот совет, установив переменную конфигурации advice.detachedHead в значение false

HEAD сейчас на 69d1f73 Добавлен новый текстовый файл

```
[ilabukanov@MacBook-Pro Kypc %
```

HEAD сейчас на 69d1f73 Добавлен новый текстовый файл

```
[ilabukanov@MacBook-Pro Kypc % git checkout master
```

Предыдущая позиция HEAD была 69d1f73 Добавлен новый текстовый файл

Переключились на ветку «master»

```
ilabukanov@MacBook-Pro Kypc %
```

GIT BRANCH



Создание новой ветки

```
[ilabukanov@MacBook-Pro Kypc % git add file.txt  
[ilabukanov@MacBook-Pro Kypc % git branch developer  
ilabukanov@MacBook-Pro Kypc %
```

```
[ilabukanov@MacBook-Pro Kypc % git checkout developer  
M      file.txt  
Переключились на ветку «developer»
```


GIT MERGE



Слияние текущей ветки с веткой developer

```
merge: developer не является тем, что можно слить  
[ilabukanov@MacBook-Pro Курс % git merge developer
```

GIT REMOTE



Устанавливаем связь с удаленным репозиторием

```
ilabukanov@MacBook-Pro Kypc % git remote add origin https://github.com/ilyabukanov123/test.git  
ilabukanov@MacBook-Pro Kypc %
```

СИСТЕМА КОНТРОЛЯ ВЕРСИЯМИ GIT



Просмотр списка подключенных удаленных репозиторий и удаление подключения

```
(используйте «git add», чтобы проиндексировать)  
[ilabukanov@MacBook-Pro Kypc % git remote  
origin  
ilabukanov@MacBook-Pro Kypc % █
```

```
[ilabukanov@MacBook-Pro Kypc % git remote rm origin2
```

GIT PUSH



Отправляем файлы с локального репозитория на удаленный

```
fatal: Authentication failed for 'https://github.com/ilyabukanov123/test2.git/'
ilabukanov@MacBook-Pro Kypc % git push -u origin master
Username for 'https://github.com': ilyabukanov123
Password for 'https://ilyabukanov123@github.com':
Перечисление объектов: 15, готово.
Подсчет объектов: 100% (15/15), готово.
При сжатии изменений используется до 8 потоков
Сжатие объектов: 100% (10/10), готово.
Запись объектов: 100% (15/15), 1.42 КиБ | 1.42 МиБ/с, готово.
Всего 15 (изменений 6), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (6/6), done.
To https://github.com/ilyabukanov123/test2.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
ilabukanov@MacBook-Pro Kypc %
```

GIT CLONE



Скачивание файлов с удаленного репозитория

```
Проект с пользователем с загрузкой фото.zip
ilabukanov@MacBook-Pro Курс % git clone https://github.com/ilyabukanov123/test2.git
Клонирование в «test2»...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 15 (delta 6), reused 15 (delta 6), pack-reused 0
Получение объектов: 100% (15/15), готово.
Определение изменений: 100% (6/6), готово.
ilabukanov@MacBook-Pro Курс %
```

```
ilabukanov@MacBook-Pro Курс % rm -rf .git
ilabukanov@MacBook-Pro Курс %
```