

ait-tr /
cohort44E

<> Code

Issues

Pull requests

Actions

Projects

Security

Insights

cohort44E / git_and_linux / lesson_02 / theory.md



dedVitalik adds git lesson_02

3 weeks ago



117 lines (87 loc) · 4.62 KB

Preview

Code

Blame

Raw



Базовые понятия

- проект - каталог с файлами исходного кода (кодовая база)
- репозиторий - хранилище истории разработки проекта
- клонирование (репо) - скачивание репо на компьютер

Базовый порядок работы

1. Инициализация нового репо

- i. `git init`
- ii. создается "скрытое хранилище" - каталог `.git/`

2. Сохранение

- Индексация файлов (добавить в очередь на сохранение)
 - a. `git add .`
- Выполнить сохранение
 - `git commit -m 'update'`

3. Привязка репо

4. Выгрузка ветки на GitHub

- i. `git push -u origin main` (main или другое название)
- ii. `git push` (если ветку уже выгружал)

Файл README.md

- использует формат Markdown
- описание репо на GitHub
- помещается в корень проекта, как правило

Пример

```
## Test Project
```



Работа с привязкой

Удалить старую привязку

```
git remote rm origin
```



Добавить привязку

```
git remote add origin скопированная_ссылка
```



Просмотр текущей привязки

```
git remote -v
```



Ветвление в Git

- Ветка - еще одна версия проекта (изолированный поток разработки)



Стратегии ветвления в Git

1. Git Flow

- i. main/master/stable - long-term (только для проверенного, оттестированного кода - "священный грааль")
- ii. develop/current - long-term (для тестирования, текущая разработка)
- iii. login/bugfix1/payments - short-term

2. GitHub Flow

- i. main/master/stable - long-term
- ii. login/bugfix1/payments - short-term

Базовые команды по работе с ветками в Git

- `git branch` просмотр веток
 - `git branch -avv` подробный вывод
 - выйти из просмотра - `q`

- `git branch новая_ветка` создать ветку
 - `git branch новая_ветка старая_ветка`
- `git checkout ветка` переключиться на ветку
 - `git checkout -` переключиться на предыдущую ветку
 - **ПЕРЕКЛЮЧАТЬСЯ НЕОБХОДИМО С "ЧИСТЫМ СТАТУСОМ"**
- `git checkout -b новая_ветка` создать и переключиться
 - `git checkout -b новая_ветка старая_ветка`
- `git branch -D ветка` удалить ветку (локально)
- `git merge название_ветки` слияние веток
- `git push origin --delete ветка` удалить ветку (дистанционную)
- `git branch -m новое_название` переименовать ветку (локально)

Слияние веток

- перенос (интеграция) изменений из одной ветки в другую
- выполняется командой
 - `git merge название_ветки`
- перед слиянием необходимо переключиться в целевую ветку

Пример

```
git checkout -b login
# внести правки
git checkout master
git merge login
git branch -D login
```



Клонирование репо

1. Открыть репо на **GitHub**
2. Решить, куда его скачать
3. Скопировать SSH-ссылку
4. Выполнить команду
 - i. `git clone скопированная_ссылка`

Ссылки

- [клонирование](#)
- [ветвление](#)
- [клонирование через IDEA](#)

