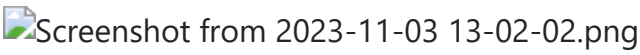
Plan

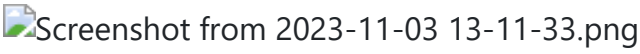
Задачи на урок:

1. Изучить основы работы с ветками в Git

Theory

!https://prod-files-secure.s3.us-west-2.amazonaws.com/95d3eea4-bdd9-4866-805a-55b03d066b78/5559ce3c-0b32-4dae-910a-d2d38aad47c2/Screenshot*from*2023-09-11_14-32-55.png


Screenshot from 2023-11-03 13-02-02.png


Screenshot from 2023-11-03 13-11-33.png

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**
    1. `git init`
    2. создается “скрытое хранилище” - каталог `.git/`
2. **Сохранение**
    - Индексация файлов (добавить в очередь на сохранение)
        1. `git add .`
    - Выполнить сохранение
        - `git commit -m 'update'`
3. **Привязка репо**
4. **Выгрузка ветки на GitHub**
    1. `git push -u origin main` (`main` или другое название)
    2. `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**
    1. `main/master/stable` - long-term (только для проверенного, оттестированного кода - “**священный грааль**”)
    2. `develop/current` - long-term (для тестирования, текущая разработка)
    3. `login/bugfix1/payments` - short-term
2. **GitHub Flow**
    1. `main/master/stable` - long-term
    2. `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
    1. `git clone скопированная_ссылка`

Ссылки

- клонирование
- ветвление
- клонирование через IDEA

Homework

Homework

Задание 1

1. Применить базовые команды по работе с ветками

Code