

Правила работы с репозиториями в MOEVM Repositories (Forgejo)

Мастер-классы по работе с git

1. [Как создать свой первый пулл реквест \(видео версия\)](#)
2. [Пошаговая инструкция по отправке работ на Forgejo для чайников](#)
3. [Как внести изменения в pull-request \(Короткая версия\)](#)

Требования к созданию Pull-Request'a

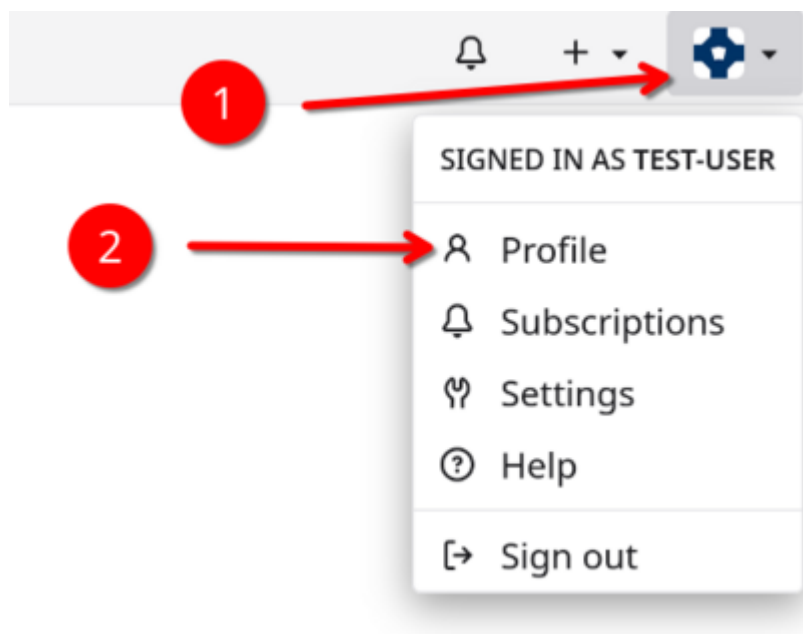
1. Настройки названия аккаунта в конфигурации

1.1. Текущие настройки конфигурации

Текущие настройки конфигурации можно вывести командой:

```
git config --list
```

В выводе этой команды необходимо обратить внимание на первые две строки: **user.name** и **user.email**. В них должны быть указаны корректные данные акканута студента. Свои данные можно найти в настройках аккаунта на git.moevm.info





1.2. Добавление данных в конфигурацию

Свои данные можно добавить следующими командами:

```
git config --local user.email ivan@ivanov.com
git config --local user.name IvanIvanov
```

, где `ivan@ivanov.com` – это почта, указанная в аккаунте IvanIvanov



Разница между `--global` и `--local` остаётся на [самостоятельно изучение](#)



Проверяйте настройки конфигурации, прежде чем делать Pull Request с компьютера в компьютерном классе на кафедре



Если в компьютерном классе использовался флаг `--global` то очистите свои данные при помощи команд из п. 1.2, чтобы следующий студент на этом компьютере не сделал свой pull-request в другой репозиторий с вашими данными.

2. Именованние ветки: <Фамилия>_<Имя>_<Вид и номер работы>

Пример именованния ветки для студента Иванова Ивана, который хочет выполнить первую лабораторную работу:

```
Ivanov_Ivan_lb1
```

Для указания лабораторных работ необходим префикс `lb`, а для курсовой работы – `sw`. При этом для курсовой работы номер работы не нужен, так как курсовая работа единственная в семестре.



Название `lr` или любое другое для лабораторной работы (как и для курсовой работы) засчитано не будет.

3. Комментарий коммита должен быть осмысленным

Для добавления сообщения к коммиту можно использовать флаг `-m`:

```
git commit -m "Ivanov LB1: fixed the corner case in X function"
```

Комментарий к коммиту должен быть осмысленным. Сообщения всех добавленных коммитов должны содержать фамилию и название работы, которая сделана в рамках коммита.

4. Каждая новая лабораторная/курсовая/контрольная работа должна находиться в своей ветке, которая обязательно должна быть создана из ветки `main`

Пример перехода на ветку `main` и создания + перехода в новую ветку `Ivanov_Ivan_lb3`:

```
# Переход на ветку main
git checkout main
# Обновлении локальной ветки main из удаленного репозитория
git pull origin main
# Создание и переход на новую ветку
git switch -c Ivanov_Ivan_lb3
```

5. Каждая лабораторная/курсовая/контрольная работа должна находиться в своей папке, которая должна называться также, как ветка

Пример создания папки на Linux:

```
mkdir Ivanov_Ivan_lb3
```

6. В репозитории должен храниться только исходный код к лабораторным и отчёты

Файлы с исходным кодом должны находиться в **папке** `src` внутри папки с лабораторной работой. Никаких других файлов не должно быть в `pull-request'e`.

Пример добавления файлов, создания коммита и отправки коммитов на удаленный

репозиторий:

```
# Добавление файла с исходным кодом
git add Ivanov_Ivan_lb3/src/main.c
# Добавление отчёта, если таковой необходим
git add Ivanov_Ivan_lb3/Ivanov_Ivan_lb3.pdf
# Создание коммита
git commit -m "Ivanov_Ivan_lb3: done"
# Отправка ветки в удаленный репозиторий
git push origin Ivanov_Ivan_lb3
```

В корне репозитория лежит файл README.md, в котором явно указаны имя, фамилия и forgejo логин студентов. Все коммиты в pull-request'e должны быть добавлены пользователем соответствующим студенту.

7. Именованние pull-request'a: <Фамилия>_<Имя>_<Вид и номер работы>

Правила для названия pull-request'a аналогичным правилам именования веток ([2. Именованние ветки: <Фамилия>_<Имя>_<Вид и номер работы>](#))

8. Все добавляемые/изменяемые/удаляемые файлы должны относиться к рабочей папке

Вне рабочей папки ничего менять нельзя, в том числе удалять чужие рабочие папки и добавлять свои документы в корень репозитория. Требования распространяется на **все коммиты** в pull-request'e. Если **хотя бы один коммит** содержит изменение вне рабочей папки, то pull-request считается некорректным. Недостаточно проверять, что **суммарные изменения** в pull-request'e не затрагивают изменения вне папки, так как отдельные коммиты могут отменять действия друг друга.

Автоматическая проверка pull-request'ов

Каждый открытый pull-request автоматически проверяется на соответствие правилам. В случае, если pull-request содержит нарушения, он закрывается с соответствующим комментарием и на него выставляется метка failed. Дополнительные действия, за которые pull-request будет **автоматически закрыт**:

1. Добавление и снятие меток с pull-request'a
2. Добавление файлов с запрещенными расширениями. К запрещенным расширениям относятся .рус, .o, .exe, .out, .txt
3. Pull-request создан пользователем с forgejo-логинем, не соответствующим имени и фамилии, которые указаны в названии pull-request'a.
4. Pull-request содержит коммиты посторонних пользователей

Если проверка была начата, то она заканчивается одним из трех результатов:

- всё хорошо (passed),

- есть нарушения (failed)

В любом из случаев в комментариях к pull-request'у будет написан результат, и на pull-request будет установлена соответствующая метка (passed/failed).



Pull-request считается корректным (т.е. **за него можно получить баллы**), если присутствует метка passed и корректный исходный код.

- [Описание меток на pull-request'e \(TBD\)](#)
- [Гайд по сообщениям автоматической проверяющей системы \(TBD\)](#)

Что делать, если мой Pull Request был закрыт?

Нужно исправить причины, по которым был закрыт Ваш Pull Request, и нажать кнопку «Reopen».

Нельзя создавать новый Pull Request при наличии уже существующего с метками преподавателя или комментариями преподавателя. При нарушении данного правила баллы за соответствующую работу **могут быть аннулированы (без возможности перезащиты) или снижены** на усмотрение преподавателя по лабораторным работам.

Если по каким-то причинам не удаётся исправить предыдущий pull request, то необходимо:

1. В новом pull-request'e оставить **ссылку на предыдущий pull-request**
2. Указать в комментарии **возникшую проблему в предыдущем pull-request**
3. **Сообщить преподавателю** по лабораторным работам о новом pull-request'e (см. [Правила коммуникации](#))

Опечатка в названии аккаунта в истории коммитов -- что делать?

Часто бывает так, что при настройке *git config* случаются опечатки в названии аккаунта или почты. Любой неправильный символ приводит к тому, что система автоматической проверки распознает pull-request как сделанный некорректно.

Чтобы исправить данную ошибку необходимо:

1. Удалить ветку в удаленном репозитории
2. Удалить ветку в локальном репозитории
3. Создать ветку заново
4. Сделать новый коммит
5. Отправить ветку в удаленный репозиторий

```
# Удалить ветку в репозитории (можно сделать из веб версии)
# Настроить git config правильно
git config --local user.email ivan@ivanov.com
git config --local user.name IvanIvanov
```

```
# Переключиться на ветку main
git switch main
# Удалить ветку с неправильным коммитом
git branch -d Ivanov_Ivan_lb3
# Создать ветку
git switch -c Ivanov_Ivan_lb3
# Добавить файлы
git add Ivanov_Ivan_lb3/src/main.c
git add Ivanov_Ivan_lb3/Ivanov_Ivan_lb3.pdf
# Создать коммит
git commit -m "Ivanov_Ivan_lb3: done"
# Отправить ветку в удаленный репозиторий
git push origin Ivanov_Ivan_lb3
```

Коммиты в ветку main



Запрещено делать любые коммиты в ветку main

Это влечет за собой **минус 1 балл к итоговому рейтингу**. Штрафной балл начисляется за **каждый коммит!** Например, если сделано 3 коммита в ветку main, то к рейтингу добавляется 3 штрафных балла.

Что делать, если я случайно выполнил слияние (merge) своего pull-request'a?

- Теряются баллы за pull-request и баллы за защиту сmergedной лабораторной работы.
- Для получения баллов за pull-request **решить другой вариант лабораторной работы**
 - «Другой вариант» - следующий после того, который решался до этого
 - Предварительно необходимо проинформировать **преподавателя по лабораторным работам** (см. [Правила коммуникации](#))
 - После этого можно создать новый pull-request с другим вариантом лабораторной работы
 - В комментарии нового pull-request'a необходимо указать:
 - Номер вариант, который был до
 - Ссылку на предыдущий pull-request
 - Номер нового вариант в новом pull-request
 - Если есть занятия по расписанию до крайнего срока, то можно защищать новый pull-request с другим вариантом лабораторной работы
 - Если самостоятельное слияние pull-request'a сделано после дедлайна, то баллы за защиту лабораторной работой восстановить невозможно

From:

<https://se.moevm.pro/> - **МОЭВМ Вики** [se.moevm.pro]

Permanent link:

https://se.moevm.pro/doku.php/courses:git_rules

Last update:

