

УРОК 14. GIT FORK В КОНТЕКСТЕ PR

ЧТО ТАКОЕ GIT FORK	2
ISSUE В GITHUB	4
PULL REQUESTS В GIT И GITHUB	5
ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ	11



ЧТО ТАКОЕ GIT FORK



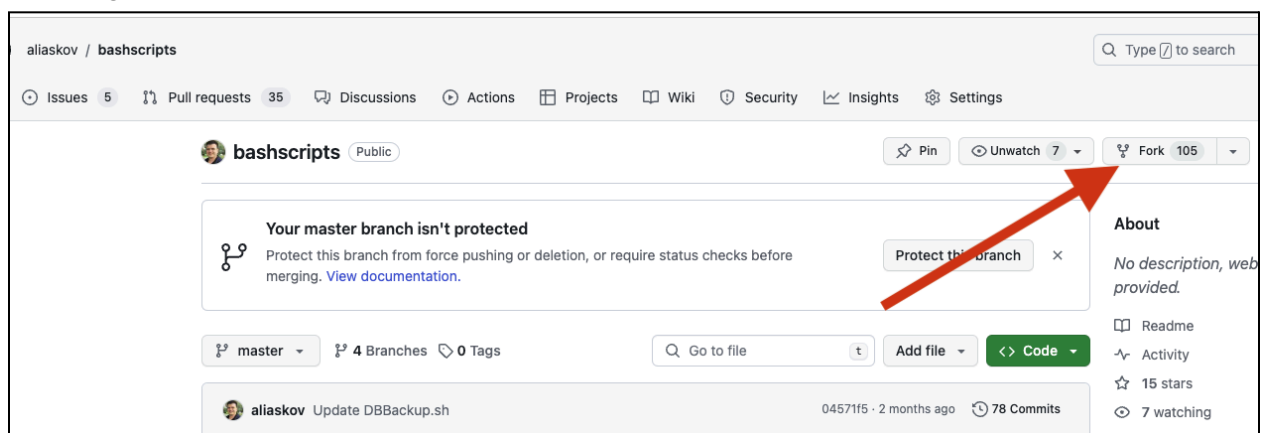
Git fork - это операция, которая создает копию удаленного репозитория на платформе Git, обычно на сервере, таком как GitHub или GitLab.

Ключевые особенности:

- Создание копии репозитория: git fork позволяет создать копию проекта на платформе Git
- Репозиторий, который Вы получаете после fork - Ваш, м;у; делать с ним что угодно. Форк позволяет вам вносить свои изменения в проект, а затем предложить их для интеграции в исходный репозиторий через pull request.
- git fork позволяет изолировать вашу работу от оригинального репозитория, обеспечивая безопасное пространство для экспериментов и изменений.
- git fork оставляет связь с оригинальным репозиторием, что позволяет обновлять его содержимое, взяв коммиты из оригинального репозитория.
- git fork и последующие Pull Requests позволяет командам и сообществам сотрудничать над проектами, предлагая изменения через Pull Request, которые могут быть рассмотрены и внесены в оригинальный проект.



Пример использования git fork:

На платформе GitHub или GitLab перейдите на страницу исходного репозитория <https://github.com/aliaskov/bashscripts> и нажмите кнопку "Fork".



После этого вы можете клонировать этот репозиторий, так как он теперь ваш, но сохраняет связь с репозиторием, по которому он был форкнут:



 Altius / **bashscripts**

Q Type [7] t

<> Code

Pull requests

Actions

Projects

Security

Insights

bashscripts Public

Watch 0 Fork

master ▾

4 Branches

0 Tags

Q Go to file

Add file ▾

<> Code ▾

About

This branch is 4 commits behind [aliaskov/bashscripts:master](#).

aliaskov connect to basti script

e7c7388 · 2 years ago

74 Commits

Readme

Activity



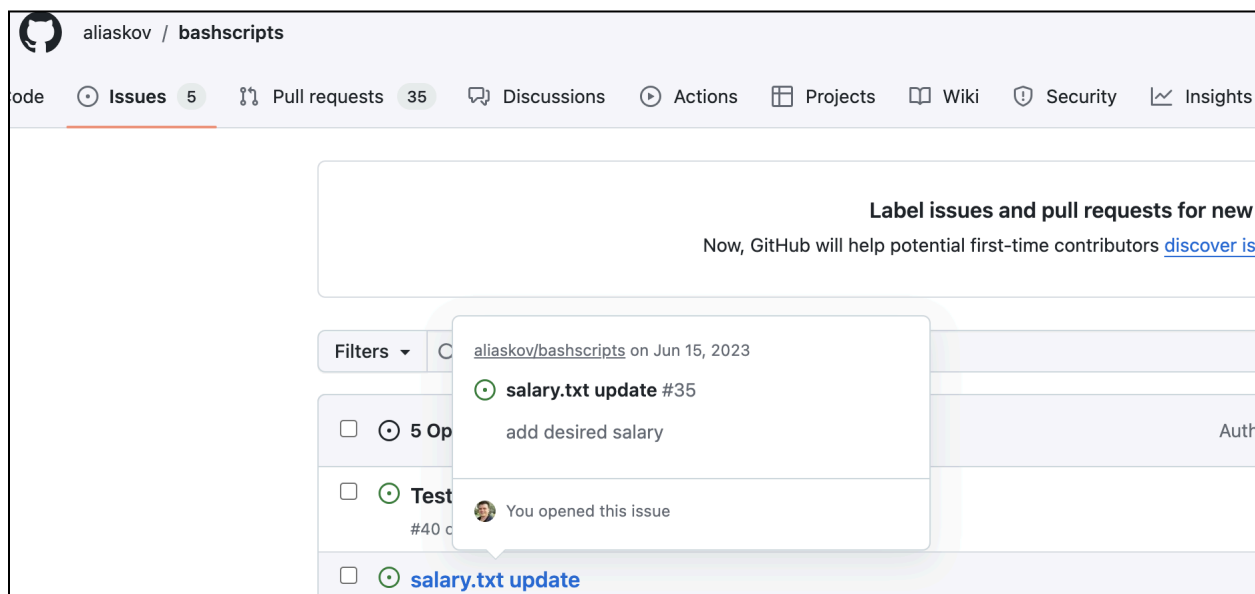
ISSUE В GITHUB

В GitHub, issue (задача) представляет собой место для обсуждения, отслеживания и управления работой над конкретной задачей или проблемой в проекте.

Каждая задача (issue) обычно представляет собой определенное изменение, исправление ошибки, предложение новой функциональности или общую проблему, требующую внимания.

Каждая задача имеет название и описание, которые помогают понять суть проблемы или предложенного изменения.

Метки используются для классификации задач по различным категориям, таким как "bug" (ошибка), "enhancement" (улучшение), "documentation" (документация) и т.д. Они помогают организовать и фильтровать задачи.



aliaskov / bashscripts

code Issues 5 Pull requests 35 Discussions Actions Projects Wiki Security Insights

Label issues and pull requests for new
Now, GitHub will help potential first-time contributors [discover is](#)

Filters on Jun 15, 2023

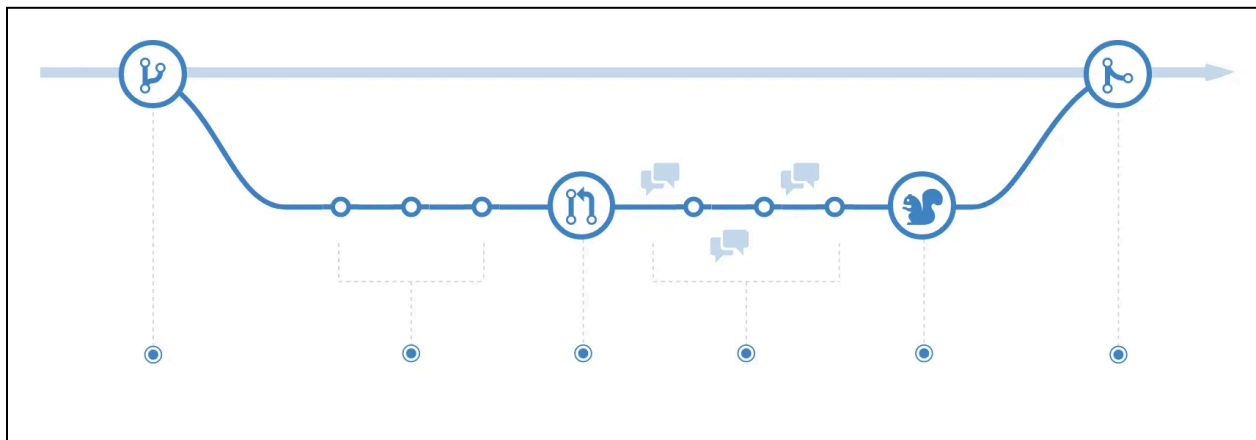
- ☐ ☒ 5 Open salary.txt update #35
add desired salary Auth
- ☐ ☒ Test #40 c You opened this issue
- ☐ ☒ salary.txt update



PULL REQUESTS В GIT И GITHUB



Pull Request - это механизм, предоставляемый некоторыми платформами совместной разработки, такими как GitHub, для предложения внесения изменений из одной ветки или форка в другую ветку или репозиторий.



Рабочий процесс pull request:

Вытащить изменения на ваш компьютер (получить самую последнюю версию изменений).

1. Создать "ветку" (версию).
2. Зафиксировать изменения. (закоммитить)
3. а Загрузить (запустить) ваши изменения.
б Открыть "pull request" (предложить изменения).
4. Обсудить и прорецензировать ваш код.
5. Произвести ребейз и тестирование.
6. "Слить" (смержить) вашу ветку с основной веткой.

Теперь подробнее по шагам:

1. Создать "ветку" (версию).

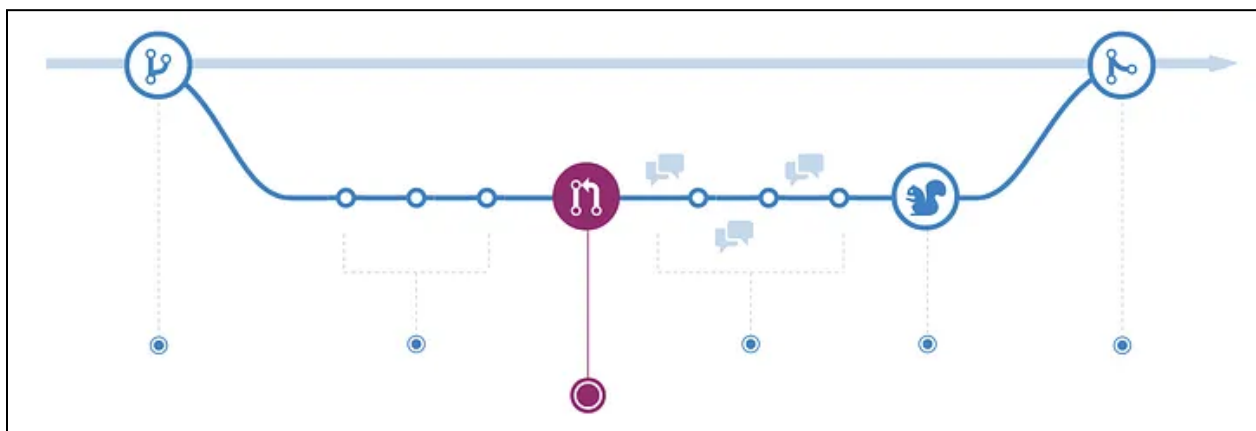
Совет: Сообщения коммитов важны, они описывают историю вашего кода. Написание понятных сообщений коммитов может сделать процесс восприятия вашего кода более простым для других людей и позволит им дать обратную связь.

3.a Загрузить (запустить) ваши изменения.

Ветка не доступна другим пользователям, пока вы не загрузите её в ваш удаленный репозиторий:

`git push origin <branch>`

3.b Открыть Pull Request

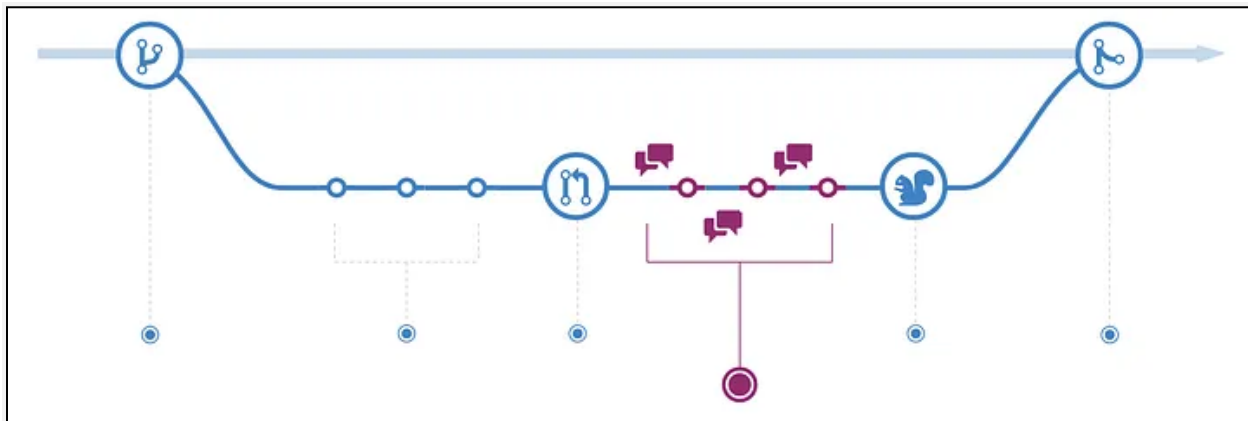


Pull Request инициирует обсуждение ваших коммитов.

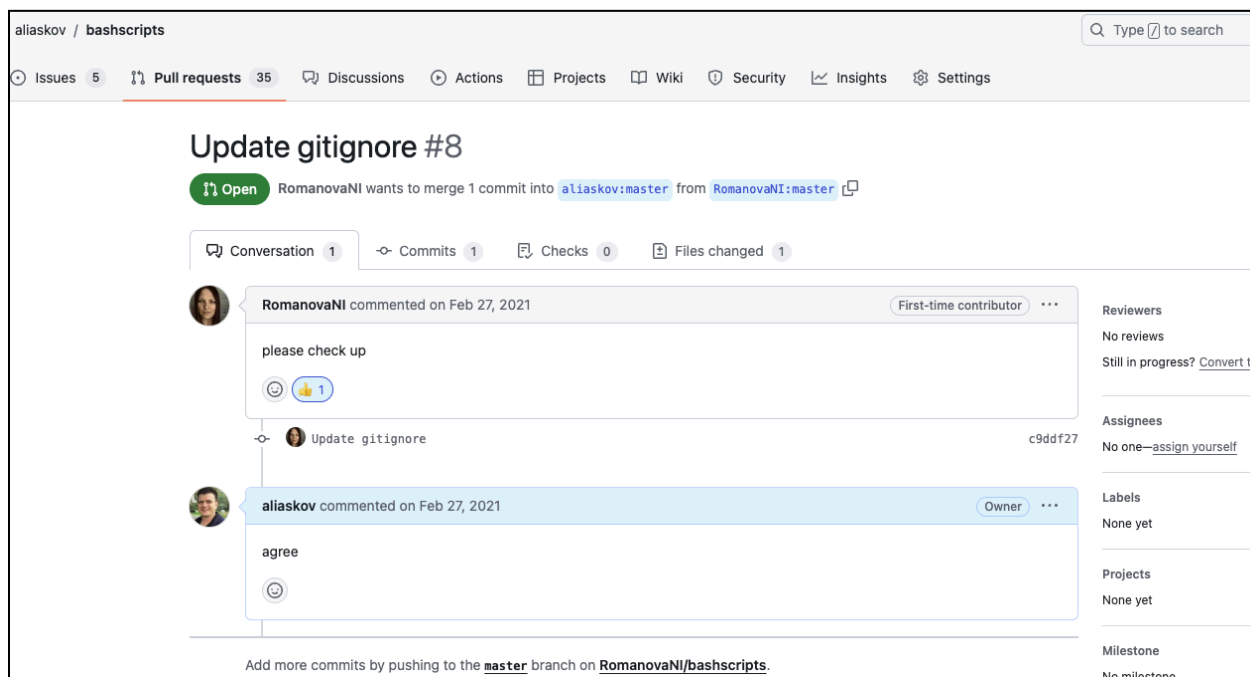
Вы можете открыть Pull Request в любой момент:

- когда у вас мало или вообще нет кода, но вы хотите поделиться скриншотами или общими идеями,
- когда вы застряли и нуждаетесь в помощи или совете,
- когда вы готовы к тому, чтобы кто-то прорецензировал вашу работу.

4. Обсудить и прорецензировать ваш код

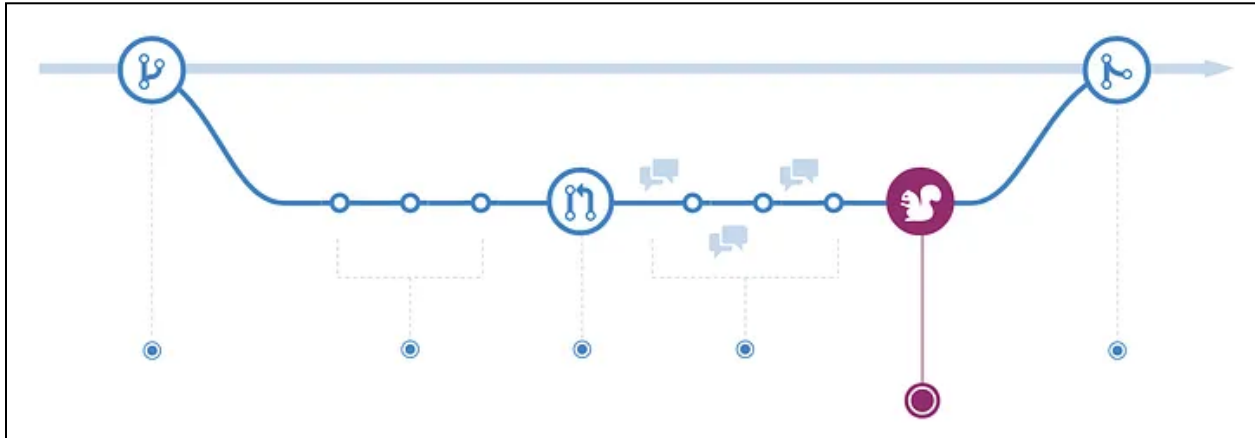


Как только Pull Request открыт, человек или команда, рецензирующая ваши изменения, может иметь возможность обсудить с автором предложенные изменения указать на конкретные строки или файлы в предложенных изменениях.



5. Ребейз и тестирование

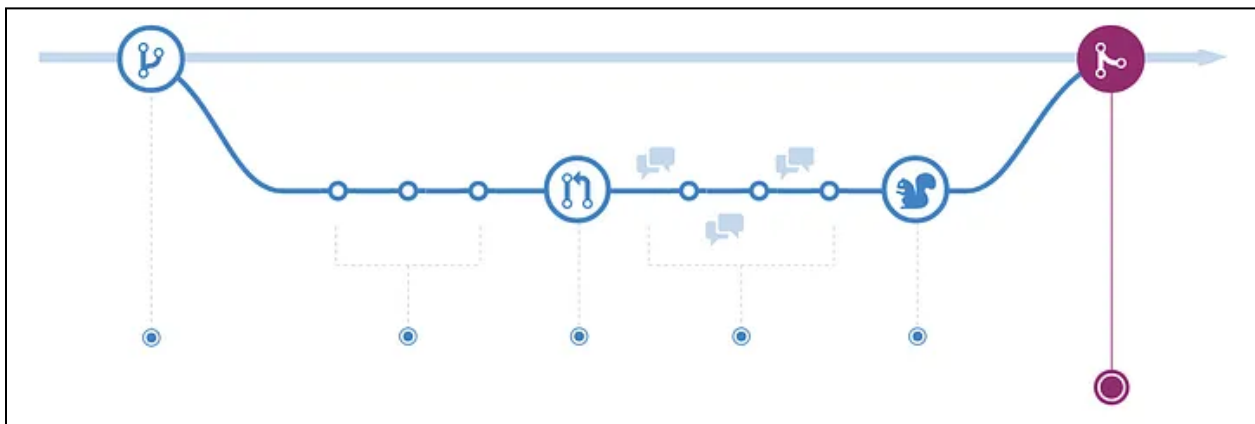
Как только ваш pull request пройдет рецензирование, а ветка пройдет ваши тесты, вы можете перебазировать (rebase) вашу ветку на master.



Чтобы воспроизвести все изменения и все коммиты на ветке master или main

Unset
`git rebase master`

6. Merge - "Слить" (смержить) вашу ветку с основной веткой.



Теперь, когда ваши изменения были проверены на продакшн, пришло время объединить ваш код с веткой master.



Unset

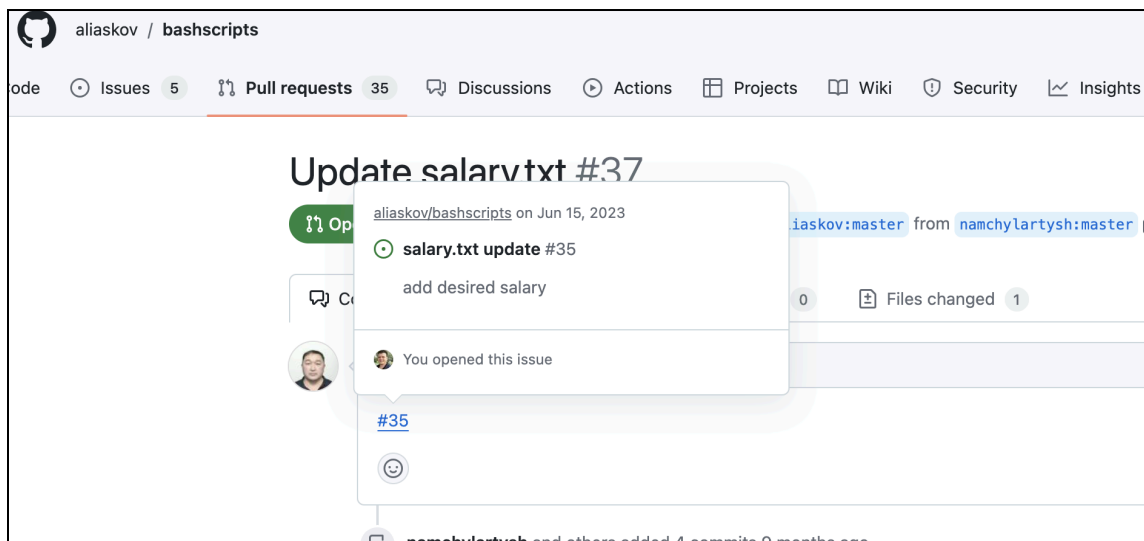
```
git checkout master
```

```
git merge <branch>
```



ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

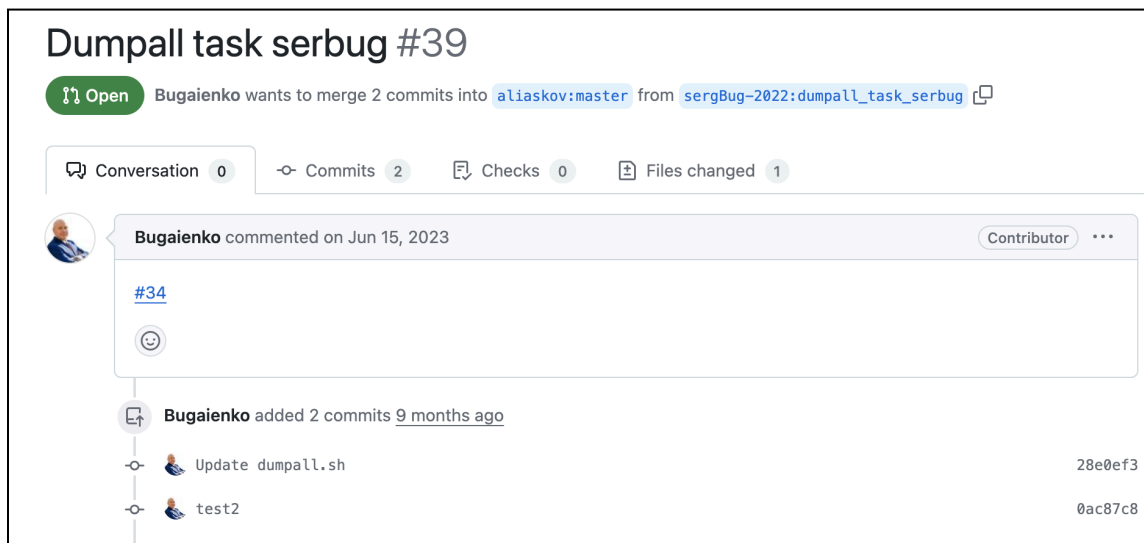
1. Создайте issue в репозитории преподавателя или того, кто будет показывать экран в процессе занятия. Опишите проблему или задачу.
2. Форкните репозиторий преподавателя или того, кто будет показывать экран в процессе занятия.
3. Внесите изменения в своей копии репозитория (например, добавьте файлы или измените изменения в файлы - добавив дату в README).
4. Дополнительно для практики веток в git - переключитесь на новую ветку, отличную от main или master.
5. Откройте PR в оригинальный репозиторий преподавателя или того, кто будет показывать экран в процессе занятия из ветки main или т.н. Feature branch, которую вы создали ранее. Не забудьте сослаться на issue (#HOMEPR_ISSUE), который вы пытаетесь решить.



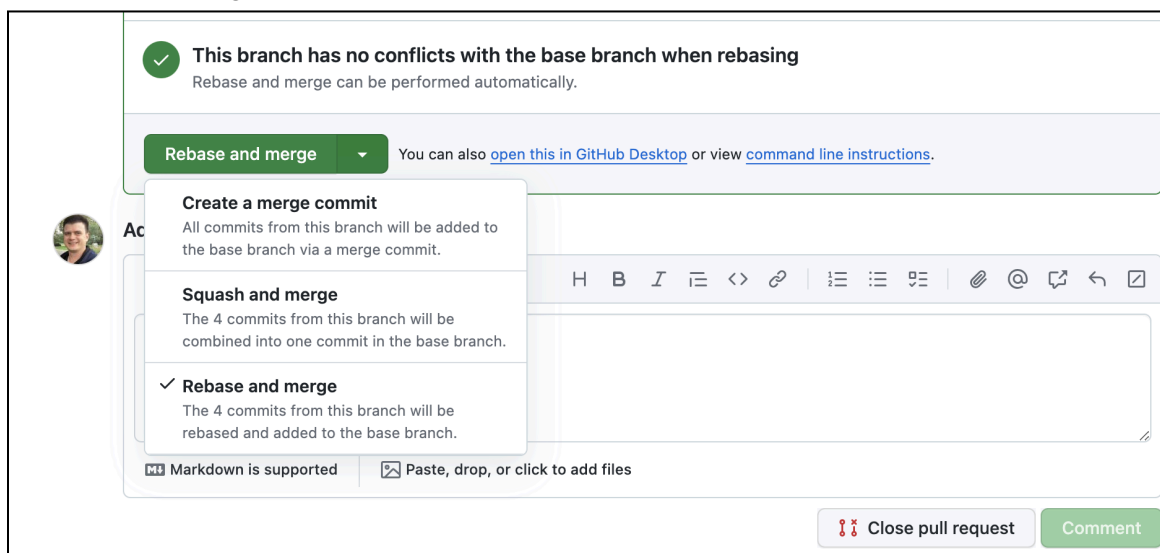
6. Обратите внимание на то, как выглядит новый PR со стороны владельца репозитория.
7. Отработайте сценарий, когда владелец репозитория, в котором открыт PR, не согласен с некоторыми файлами и строками кода в предложенном PR, попробуйте функционал уведомлений и чата в открытом PR. Обратите внимание на динамические уведомления.



- Внесите изменения в локальных репозиториях, выполните коммит изменений (или несколько) и убедитесь, что они появятся в открытом PR, новый открывать не нужно (только если открытый PR не был закрыт).



- Выполните merge этого PR. Рассмотрите функционал Rebase and merge, Squash and merge.



- Выполните pull в репозиториях и выполните merge следующего PR.