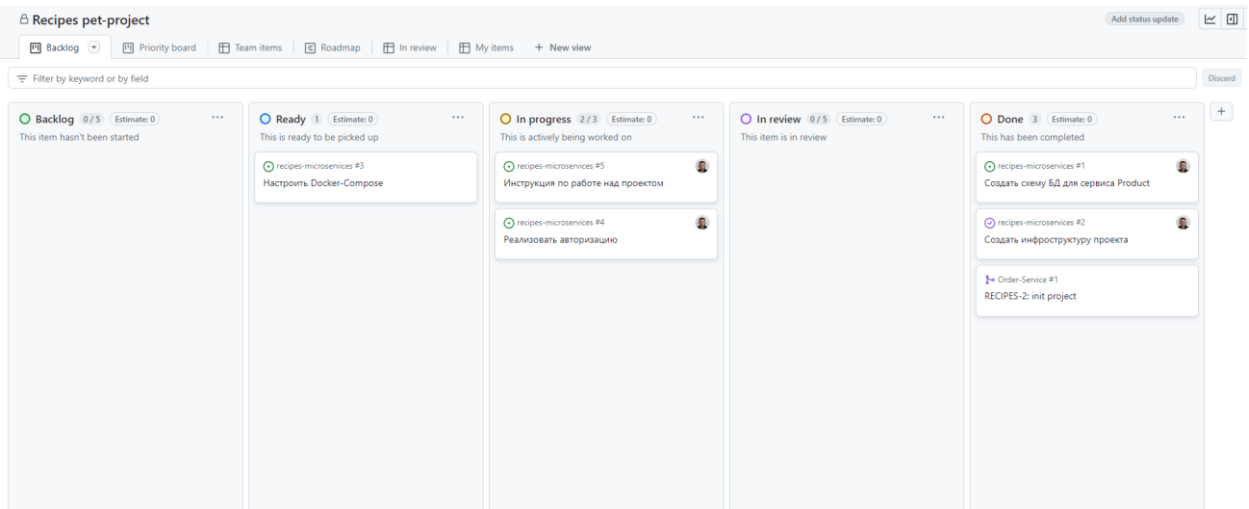


Работа с доской задач

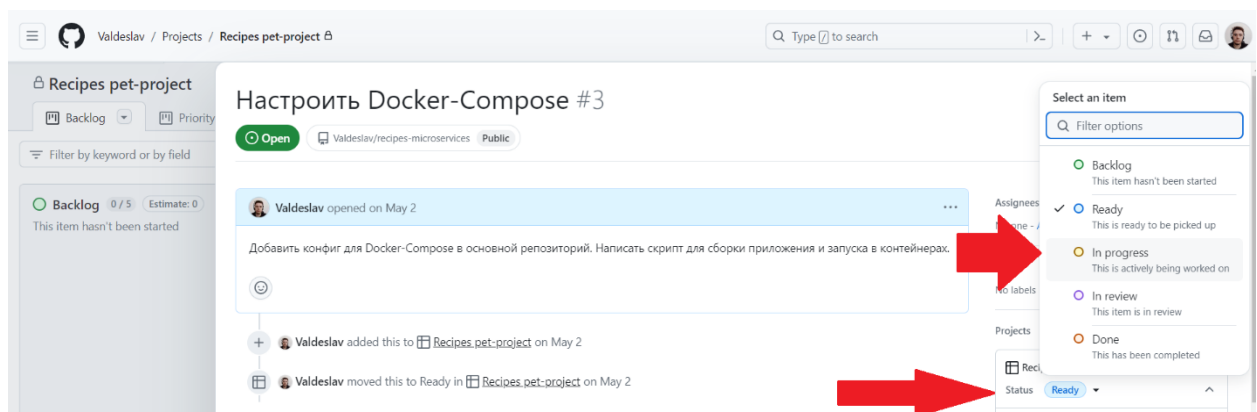
Доска задач разделена на 5 частей:



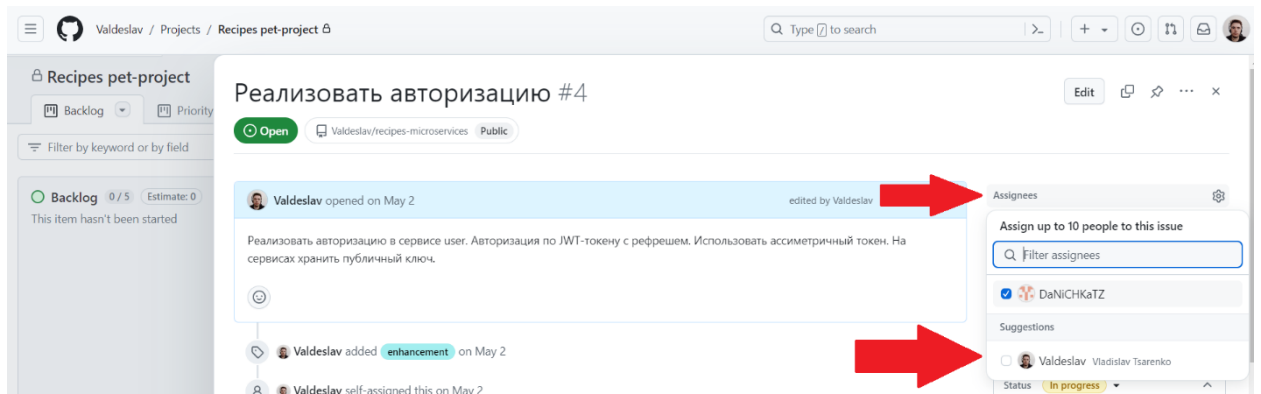
- **Backlog** – задачи над которыми не ведется работа (некорректная постановка и т.п.)
- **Ready** – задачи, готовые к выполнению.
- **In progress** – задачи, над которыми ведется работа.
- **In review** – задачи, которые находятся на проверке.
- **Done** – успешно выполненные, либо закрытые за ненадобностью задачи.

Новые задачи попадают в группу *Ready*. Оттуда их можно забрать на выполнение. Как правило, исполнитель задачи будет назначен мной при создании. Либо можно брать в работу задачи, где исполнитель не указан.

Если задача в работе, она переводится в статус *In Progress*.



После выполнения задачи, статус переводится в *In Review*, при необходимости создается pull request. В Assignees нужно добавить меня (Valdeslav).



В отдельных случаях задачу можно переводить сразу в статус *done*.

После проверки, я перевожу задачу в статус *Done*, если она выполнена успешно. Если будут какие то недочеты в реализации, я верну задачу в статус *Ready*. В комментариях к пулл реквесту можно будет посмотреть свои недочеты, и исправить их. В некоторых случаях комментарии могут быть оставлены и в самой задаче.

После исправления задача отправляется снова в статус *In review*.

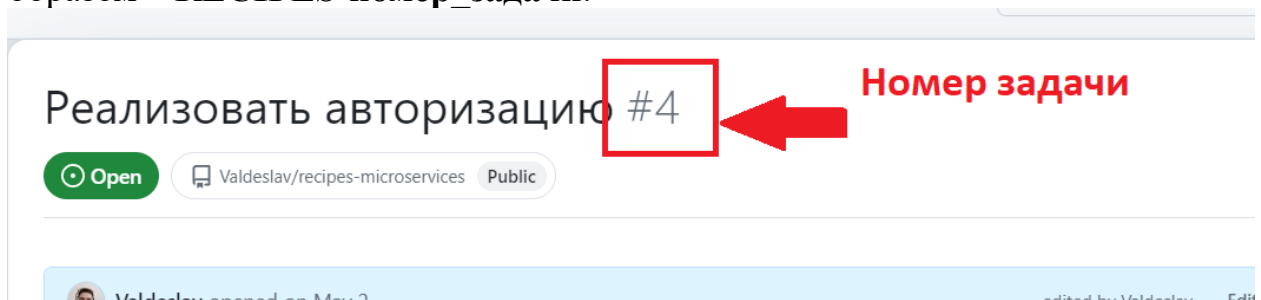
Работа с Git

В описании задачи буду указывать ссылку на репозиторий, в котором ее нужно реализовать.

Первым делом нужно клонировать репозиторий к себе на компьютер (если это не было сделано ранее). Для этого используем команду:

git clone “url репозитория”.

Для каждой задачи создается отдельная ветка. Ветки именуем следующим образом – **RECIPES-номер_задачи**.



Создаем ветку командой

git branch RECIPES-4

Затем переключаемся на нее командой

git checkout RECIPES-4

Чтобы не выполнять 2 команды, можно сразу создать и переключиться на ветку командой

git checkout -b RECIPES-4

В случае если вы уже выполняли клонирование проекта, то перед созданием новой ветки вам нужно переключиться на ветку **main** командой

git checkout main

И обновить свою ветку свежими изменениями с сервера командой

git fetch

После этого алгоритм такой же, создаем новую ветку как было описано выше.

Затем вносятся изменения в нужные файлы и делается коммит. Добавить файлы в коммит можно командой

git add <имя файла>

Как правило, я меняю только нужные файлы, и затем добавляю их командой

git add .

Точка добавит все файлы в текущем каталоге. Так что если выполнять команду из корня репозитория, в коммит будут добавлены все измененные файлы.

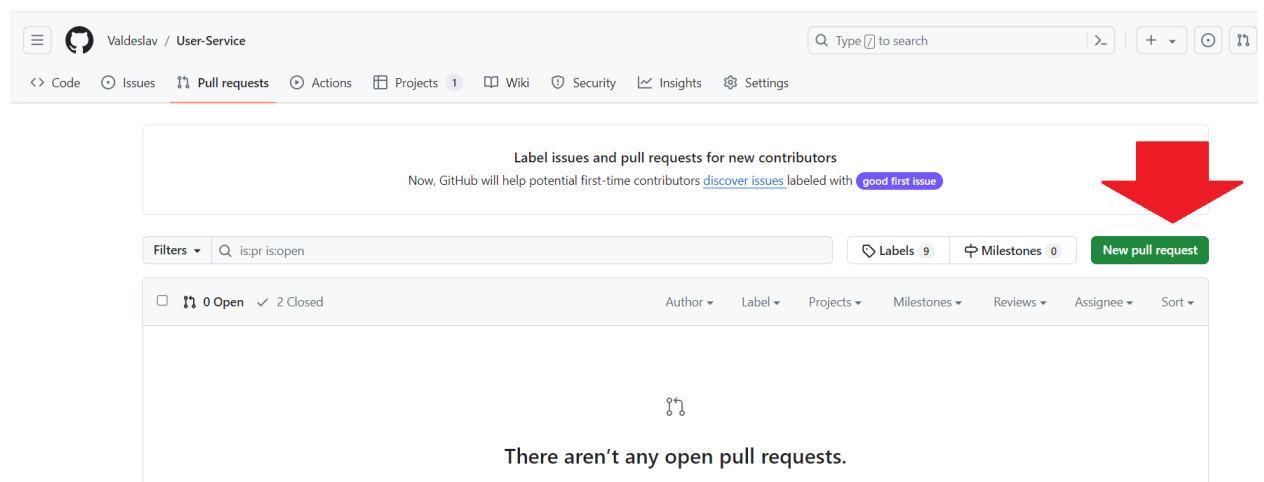
Затем делаем коммит командой

git commit -m “сообщение о коммите”

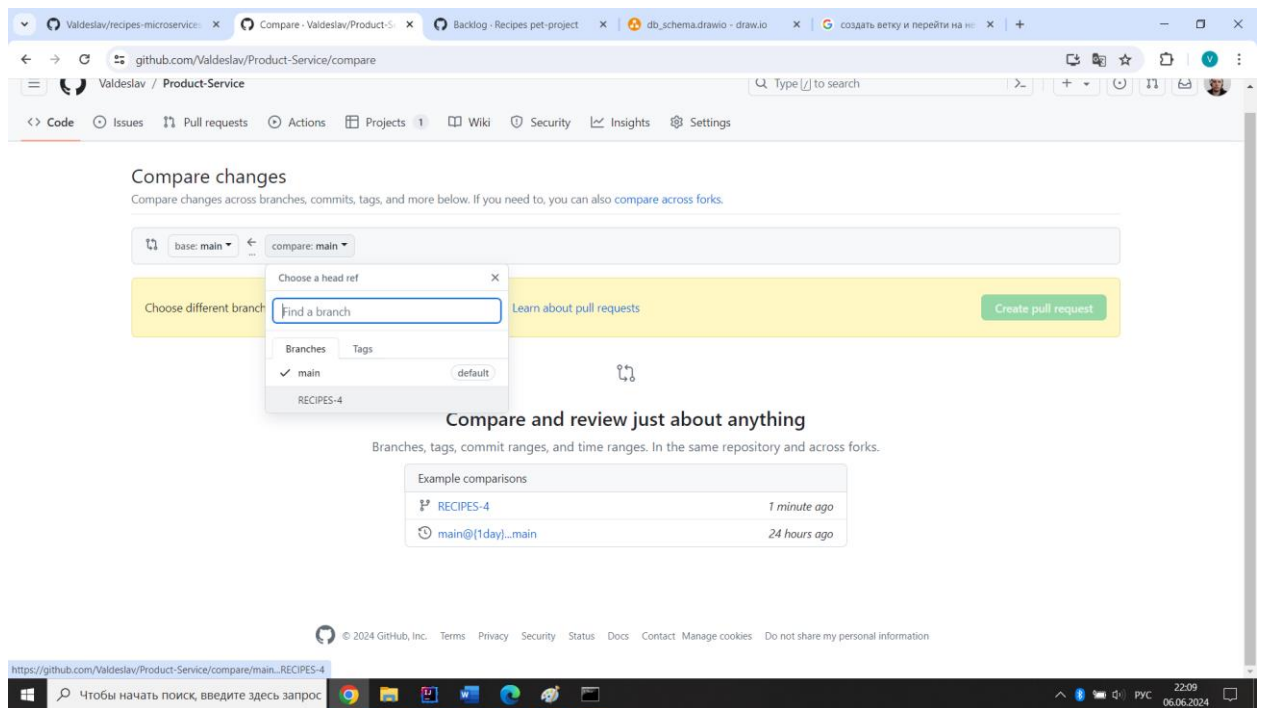
После этого, ветку нужно отправить ветку на удаленный репозиторий, чтобы другие пользователи могли увидеть ее, и была возможность сделать пулл реквест. Делается это командой

git push

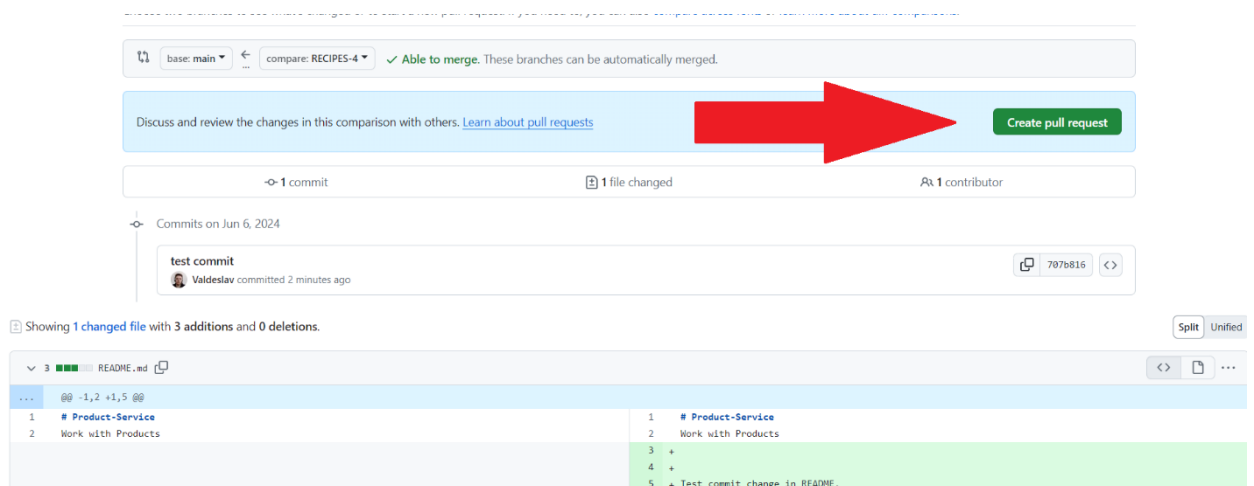
После этого заходим на страницу репозитория в гитхабе, и переходим на вкладку Pull requests. Нажимаем кнопку **New Pull Request**



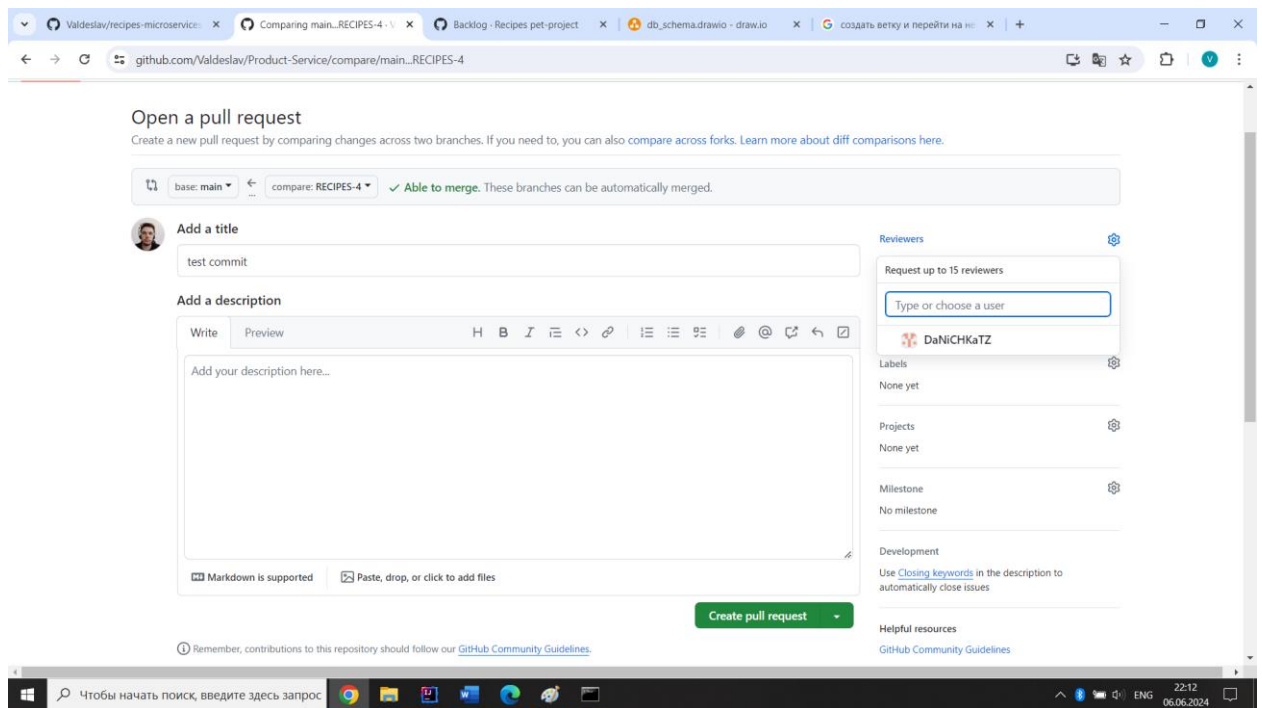
В Base оставляем ветку main, а в compare выбираем свою ветку.



Нажимаем Create pull request.



В **Reviewers** выбираем меня (Valdeslav) и жмем Create pull request. Копируем ссылку на пулл реквест, и прикрепляем ее в комментарии к задаче. Задачу переводим в статус In review.



Если после ревью нужно будет внести какие-либо изменения в код, то просто делаем коммит в той же ветке, и push. Изменения автоматически попадут в уже созданный пулл реквест.

Для удобства работы с гитом можно использовать IDE либо другие UI программы.

Здесь я описал базовый алгоритм работы с гитом. Не описал как исправлять конфликты в пулл реквесте, что делать если не удастся запустить или запулить, когда изменения на сервере расходятся с локальными. В случае возникновения проблем с гитом, гуглить либо писать мне. Многие проблемы можно решить разными способами, я расскажу про самые простые и эффективные на мой взгляд решения.

Если возникнут вопросы по работе с доской, то тоже можно спросить меня. Я мог и не учесть всех аспектов.