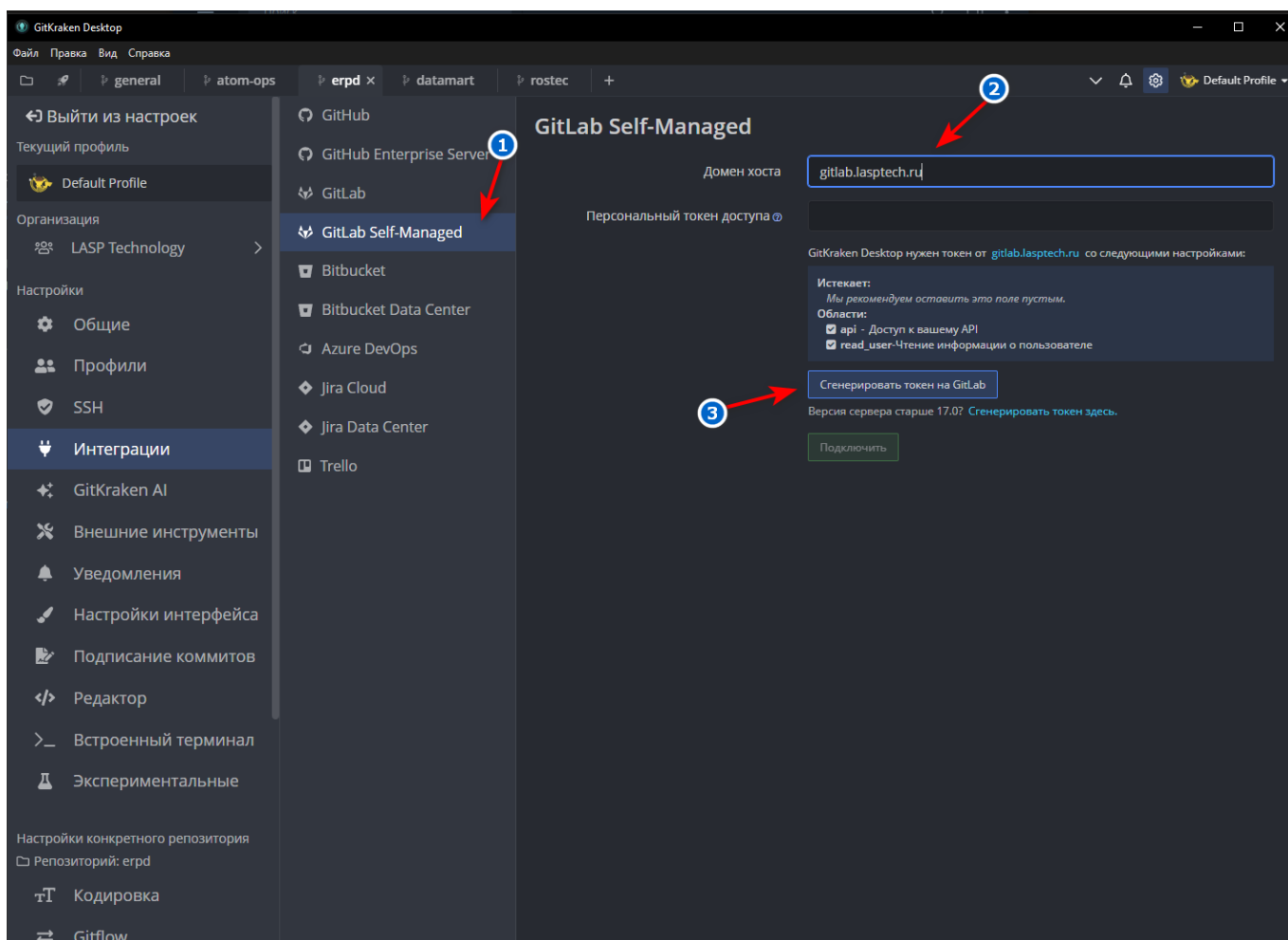


Инструкции

Переезд на свой GitLab

Выбор self-managed в GitKraken

Настройки GitKraken:



1. Выбираем **Self-hosted**.
2. Вписываем домен `gitlab.lasptech.ru`.
3. Давим `Сгенерировать токен на GitLab`.

Переход на страницу <https://gitlab.lasptech.ru>.

Создание токена



Для каждой машины на которой работаете с git'ом, можно создать персональный токен.

Настройки пользователя / Личные токены доступа

Личные токены доступа

Вы можете создать личный токен доступа для каждого приложения, которое использует GitLab API. Также можно использовать личные токены доступа для аутентификации через Git по HTTP. Это единственный допустимый пароль, если у вас включена двухфакторная аутентификация (2ФА).

Активные токены
0
Фильтровать список

Токены, срок действия которых истекает в течение 2 недель
0
Фильтровать список

Отозванные токены
0
Фильтровать список

Истёкшие токены
0
Фильтровать список

Состояние = Активный X

Срок действия

Название	Описание	Статус
Нет токенов доступа		

Generate token

Fine-grained token
Limit scope to specific groups and projects and fine-grained permissions to resources.

Legacy token
Scoped to all groups and projects with broad permissions to resources.

Токен канала

Ваш токен канала идентифицирует вас, когда программа чтения RSS загружает персонализированный канал RSS, или когда приложение календаря загружает ваш персонализированный календарь. Его можно увидеть в URL этих каналов. Его нельзя использовать для доступа к любым другим данным.

Токен канала

Держите этот токен в секрете. Каждый, кто его получит, сможет читать RSS каналы активности и задач, а также ваш канал календаря, как если бы они были вами. Если это случится, сбросьте этот токен.

Токен входящей электронной почты

Токен входящей электронной почты аутентифицирует вас, когда вы создаете новое обсуждение по электронной почте и включается в ваши личные адреса электронной почты для конкретного проекта. Его нельзя использовать для доступа к любым другим данным.

Токен входящей электронной почты

Храните этот токен в секрете. Любой, у кого он есть, может создавать обсуждения, как если бы это были вы. Если это всё-таки произошло, обновите токен.

Выбираем Legacy token.



Обязательно сбрасываем дату истечения токена.

Настройки пользователя / Личные токены доступа / **Generate legacy token**

Generate legacy token

Legacy personal access tokens are scoped to all groups and projects with broad permissions to resources.

Название токена

Описание (необязательно)

Срок действия

2026-09-02 

Выберите области действия

Области действия определяют уровни разрешений, предоставляемые токenu. Подробнее.

- ☐ **read_user**
Grants read-only access to your profile through the /user API endpoint, which includes username, public email, and full name. Also grants access to read-only API endpoints under /users.
- ☐ **read_repository**
Grants read-only access to repositories on private projects using Git-over-HTTP or the Repository Files API.
- ☐ **read_virtual_registry**
Grants read-only access to container images through the dependency proxy in private projects.
- ☐ **read_registry**
Grants read-only access to container registry images on private projects.
- ☐ **read_api**
Grants read access to the API, including all groups and projects, the container registry, and the package registry.
- ☐ **self_rotate**
Grants permission for token to rotate itself.
- ☐ **write_repository**
Grants read-write access to repositories on private projects using Git-over-HTTP (not using the API).
- ☐ **write_virtual_registry**
Grants read, write, and delete access to container images through the dependency proxy in private projects.
- ☐ **write_registry**
Grants write access to container registry images on private projects. You need both read and write access to push images.
- ☐ **api**
Grants complete read/write access to the API, including all groups and projects, the container registry, the dependency proxy, and the package registry.
- ☐ **ai_features**
Grants access to GitLab Duo related API endpoints.
- ☐ **create_runner**
Grants create access to the runners.
- ☐ **manage_runner**
Grants access to manage the runners.
- ☐ **k8s_proxy**
Grants permission to perform Kubernetes API calls using the agent for Kubernetes.

Generate token **Отмена**

Пишем уникальное и понятное название токена.

Ставим флажки как указано на картинке.

Настройки пользователя / Личные токены доступа / **Generate legacy token**

Generate legacy token

Legacy personal access tokens are scoped to all groups and projects with broad permissions to resources.

Название токена

GitKraken

Описание (необязательно)

Срок действия

YYYY-MM-DD

Выберите области действия

Области действия определяют уровни разрешений, предоставляемые токenu. Подробнее.

- ☒ **read_user**
Grants read-only access to your profile through the /user API endpoint, which includes username, public email, and full name. Also grants access to read-only API endpoints under /users.
- ☐ **read_repository**
Grants read-only access to repositories on private projects using Git-over-HTTP or the Repository Files API.
- ☐ **read_virtual_registry**
Grants read-only access to container images through the dependency proxy in private projects.
- ☐ **read_registry**
Grants read-only access to container registry images on private projects.
- ☐ **read_api**
Grants read access to the API, including all groups and projects, the container registry, and the package registry.
- ☐ **self_rotate**
Grants permission for token to rotate itself.
- ☐ **write_repository**
Grants read-write access to repositories on private projects using Git-over-HTTP (not using the API).
- ☐ **write_virtual_registry**
Grants read, write, and delete access to container images through the dependency proxy in private projects.
- ☐ **write_registry**
Grants write access to container registry images on private projects. You need both read and write access to push images.
- ☒ **api**
Grants complete read/write access to the API, including all groups and projects, the container registry, the dependency proxy, and the package registry.
- ☐ **ai_features**
Grants access to GitLab Duo related API endpoints.
- ☐ **create_runner**
Grants create access to the runners.
- ☐ **manage_runner**
Grants access to manage the runners.
- ☐ **k8s_proxy**
Grants permission to perform Kubernetes API calls using the agent for Kubernetes.

Generate token Отмена

Давим .

Токен создан. Его нужно вписать в GitKraken.

Копируем для вставки в GitKraken.

Настройки пользователя / Личные токены доступа / **Generate legacy token**

Your new token has been created

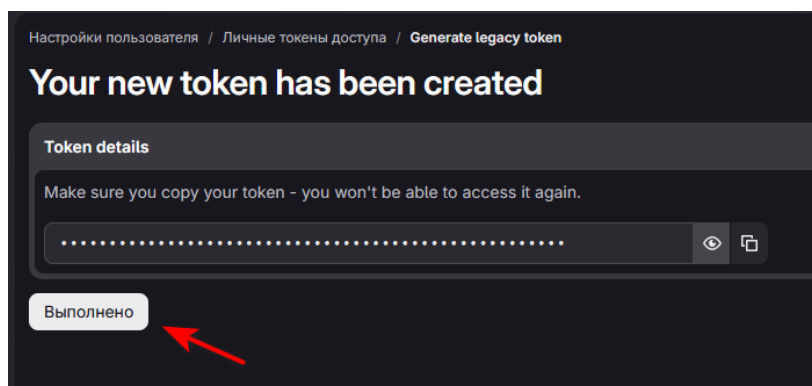
Token details

Make sure you copy your token - you won't be able to access it again.

.....

Выполнено

Давим Выполнено.

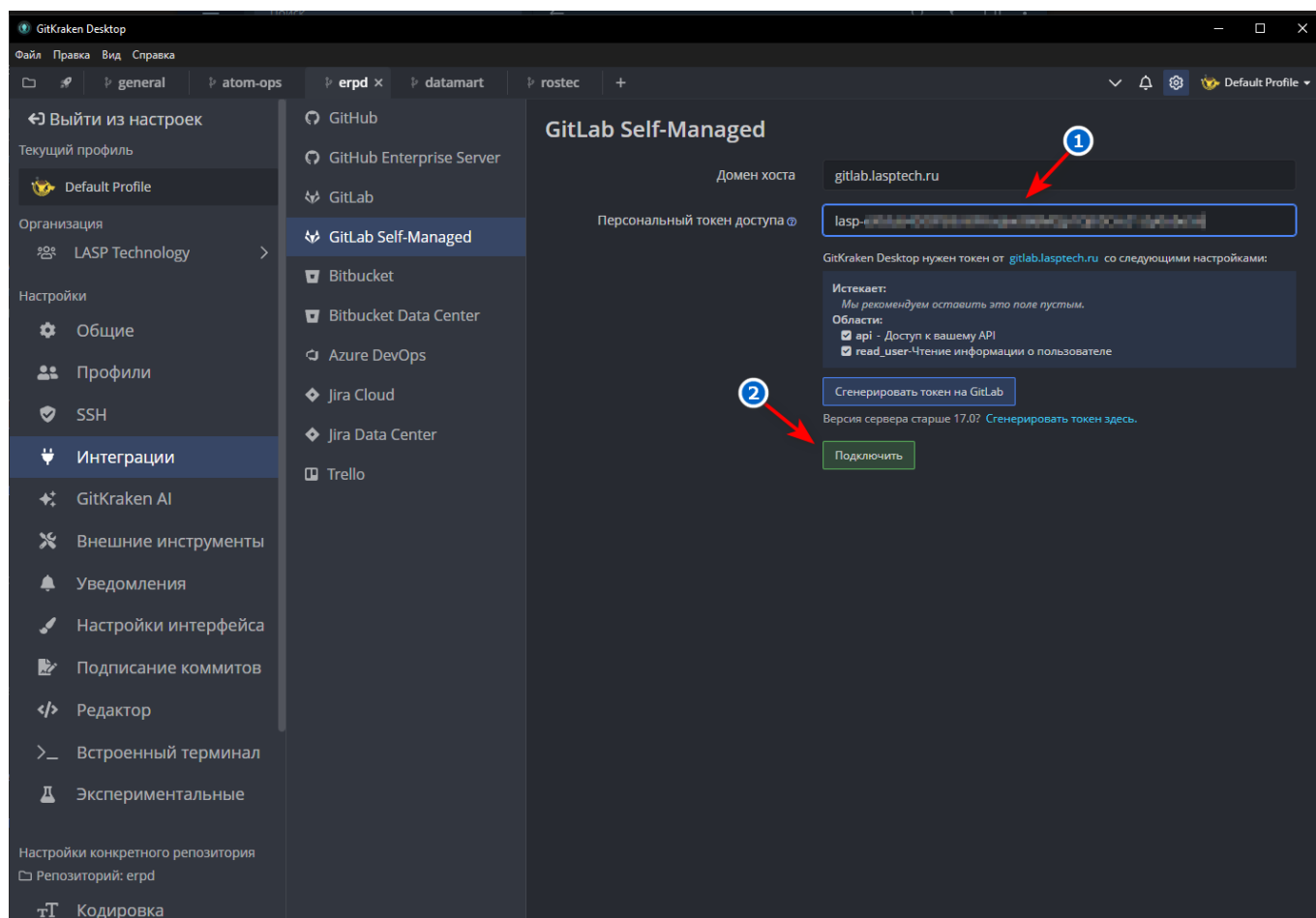


После этого токен будет невозможно посмотреть и скопировать.



Если токен каким-то образом потеряется, его нужно будет отозвать и создать новый.

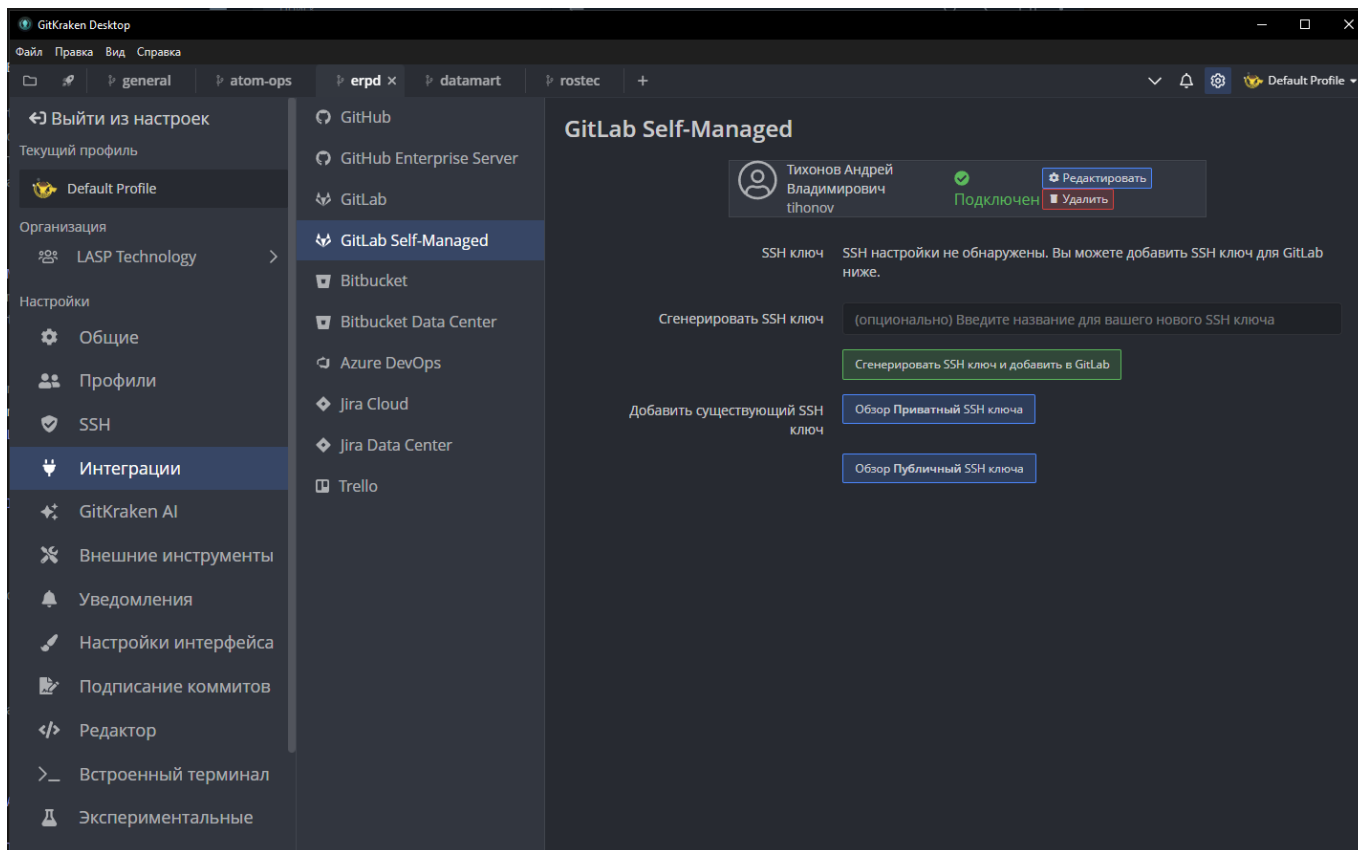
Прописывание токена в GitKraken



Вставляем скопированный токен в GitKraken.

Давим Подключить

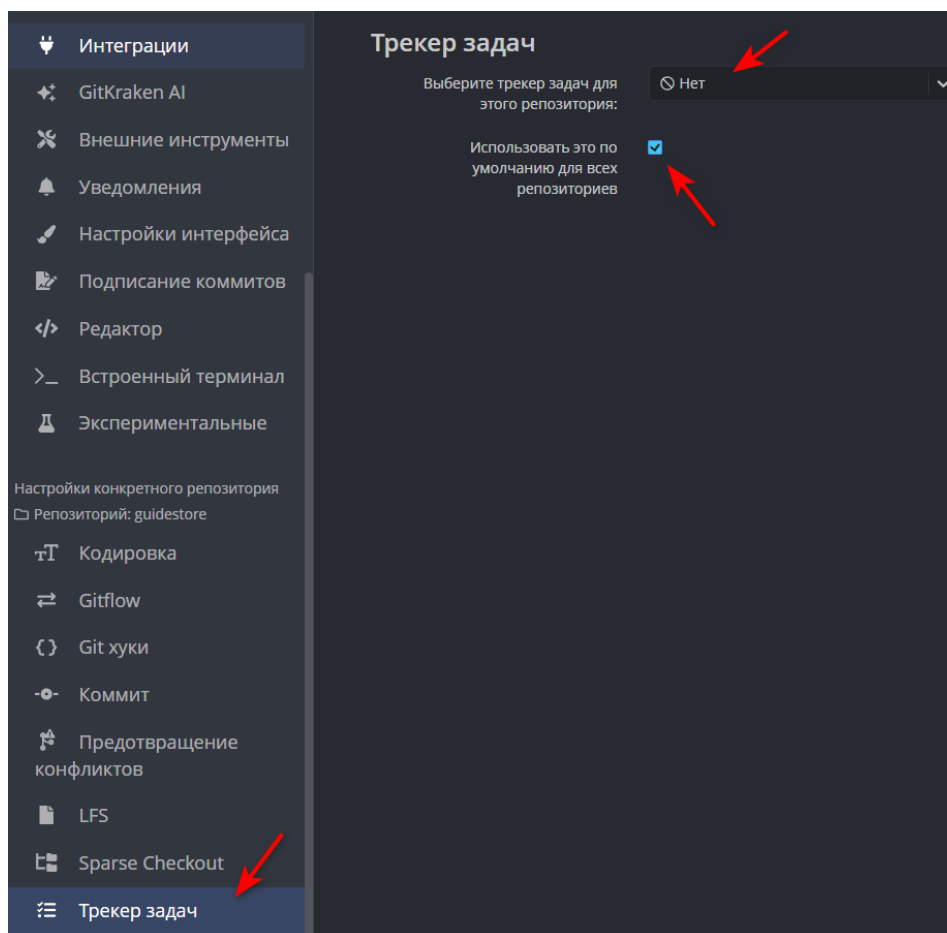
Результат:



Старую интеграцию GitLab можно отключить.

Отключение трекера задач GitLab

Тем кому трекер GitLab'a не нужен, в настройках GitKraken сделайте это:



Перейти в **Трекер задач** GitKraken.

Установить значение **Нет**.

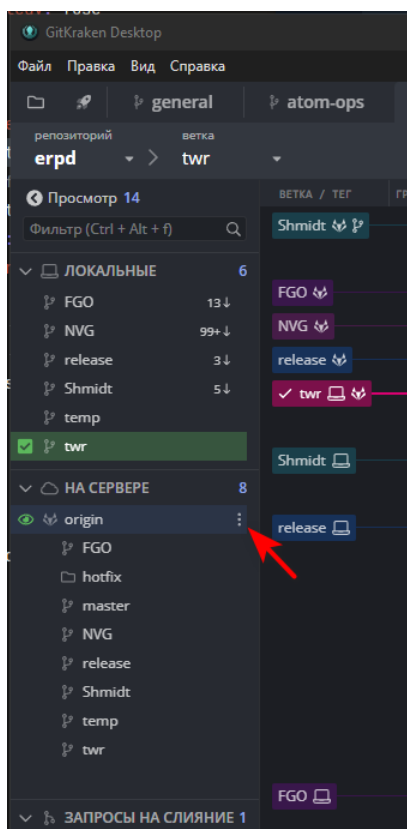
Установить флаг **Использовать по умолчанию для всех репозиториях**.



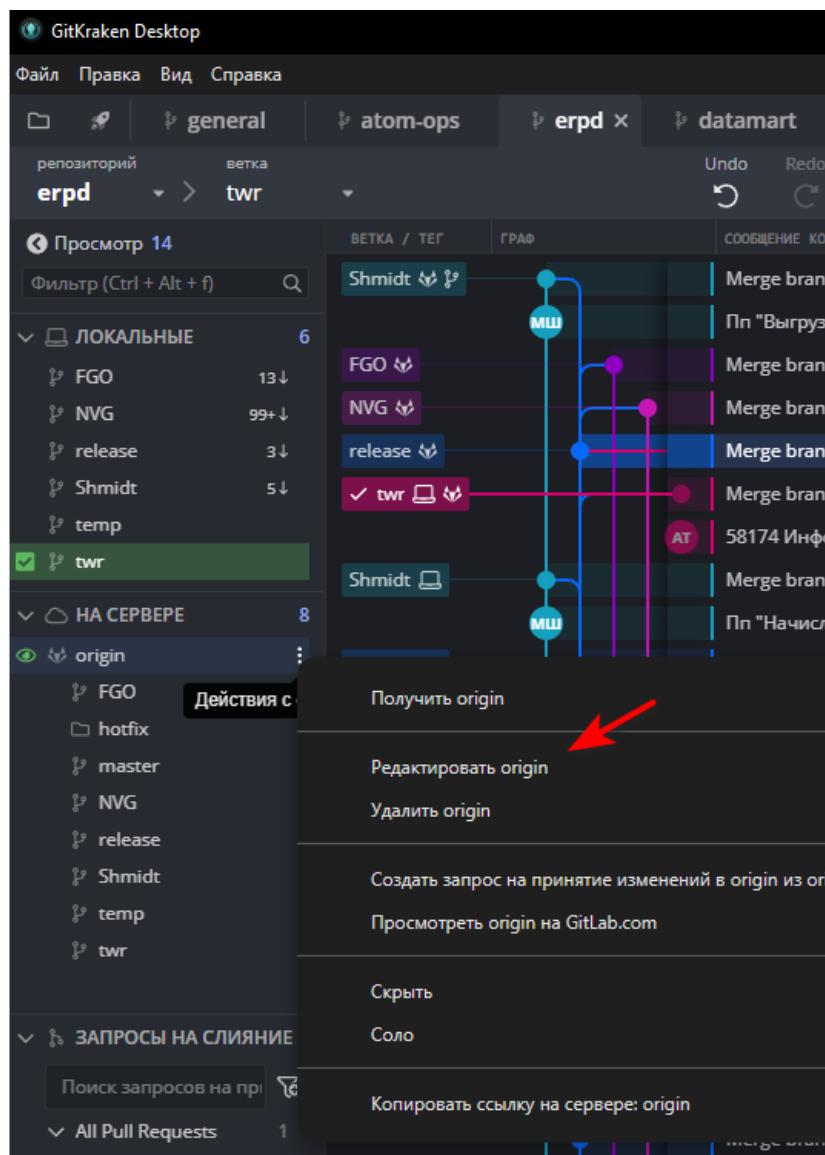
Если пользуетесь, не отключайте и настраивайте самостоятельно.

Редактирование Origin

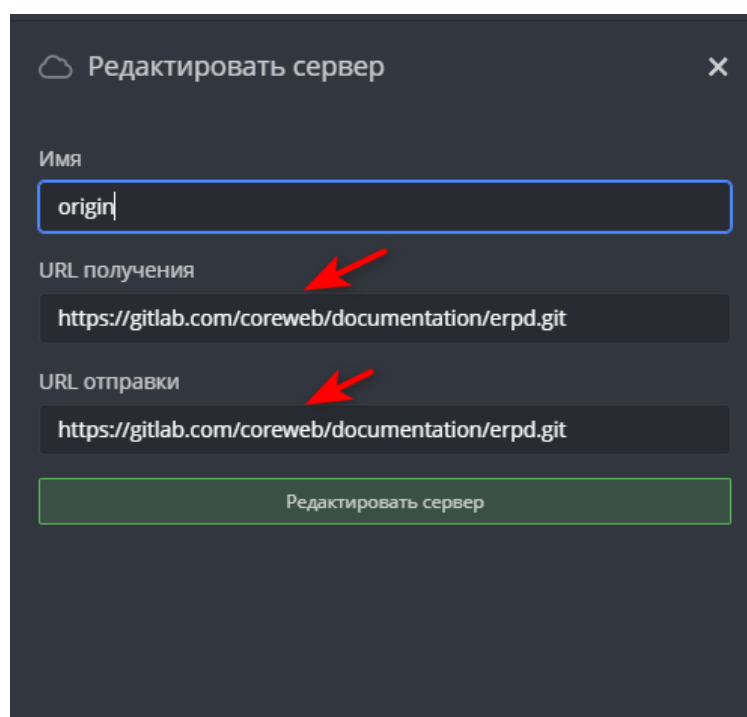
Давим на точки *origin*...



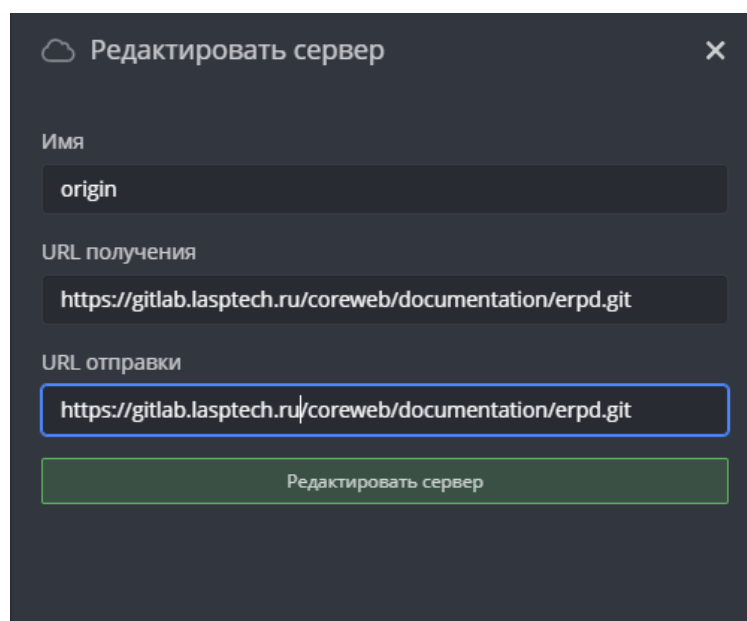
В меню выбираем *Редактировать origin*...



Вот эти адреса нам нужно отредактировать:



Должно быть так:



Редактировать сервер

Имя

origin

URL получения

https://gitlab.lasptech.ru/coreweb/documentation/erpd.git

URL отправки

https://gitlab.lasptech.ru/coreweb/documentation/erpd.git

Редактировать сервер

Редактируем строки и давим [Редактировать сервер](#)



Так нужно сделать во всех проектах!

Завершение

Перезагружаем GitKraken и делаем тестовый коммит и пуш.