

Инструкции

Инструкция технического писателя

Конфигурация «ВСЕ»
ОИ.ВСЕ.00.00.000
v2025.11.3

История изменений

№	Дата	Основание
1	10.11.2025	Первая версия
2	14.11.2025	Заменены скриншоты и команды GitKraken в связи с русификацией.
2	19.11.2025	Добавлен порядок установки плагинов. Добавлен порядок включения partials. Добавлена структура инструкции.

Содержание

Введение	5
Про Git	6
Инструментарий	7
Подготовка к работе	8
Регистрация на GitLab	8
Авторизация GitKraken на GitLab	9
Настройка имени пользователя	12
Клонирование репозитория	13
Управление версиями	16
Начало работы	17
Фиксация изменений	17
Слияние	19
Возможные конфликты	26
Работа с изображениями	27
Настройка размера источника	27
Снятие скриншота	27
Основной редактор	31
Каталог как проект IntelliJ IDEA	31
Начальные настройки	33
Как тут все устроено	37
Порядок работы	39
Начало	39
Структура операционной инструкции	40
Редактирование текста	42
Контроль ошибок	44
Включение partials	45
Создание partials	45
Контроль ссылок	46
Включение изображений	47
Рефакторинг	49
Включение диаграмм	49
Завершение	49
Синтаксис AsciiDoc	50
Атрибуты документа	50
Оглавление	51
Заголовки	51
Основной текст	52
Параграф	52

Варианты переноса текста в параграфах (для общего понимания)	52
Тематический разрыв	53
Форматирование текста	53
Объединение в блоки	54
Принудительный разрыв страницы	55
Оформление верхних и нижних индексов	55
Клавиатурные обозначения	56
Меню (путь к операции)	56
Кнопки и иконки	57
Списки	58
Неупорядоченный список	58
Упорядоченный список	58
Упорядоченный контролируемый список	59
Смешанный список	59
Разъединение списков	60
Список описаний	60
Предупреждения (варнинги)	63
Таблицы	64
Автонумерация	65
Выравнивание колонок	66
Объединение ячеек	67
Форматирование в ячейках	68
Подписи к объектам	69
Блоки	70
Кодовый блок	70
Блок-цитата	70
Блок-пример	70
Универсальный блок	71
Ссылки	73
Изображения	73
Вложения (partials)	75
Внутренние ссылки	76
Внешние ссылки	76
Диаграммы	76

Введение

Настоящая инструкция предназначена для технических писателей создающих и редактирующих документацию по всем проектам ООО «ЛАСП Технологии» расположенным в репозитории GitLab.

Про Git

Git - это система контроля версий. Это «машина времени» для ваших работ, которая запоминает каждое изменение.

Основные понятия:

- *Репозиторий* - каталог проекта, за которым следит Git;
- *Коммит* - сохранение изменений с комментарием;
- *Отдельная ветка* - черновик для экспериментов;
- *Пуш* - отправка ваших изменений на сервер.

Зачем это нужно

- Git помнит все изменения.
- Несколько человек могут работать одновременно.
- Если что-то сломали, можно вернуться к любой удачной версии.

Инструментарий

- Редактор IntelliJ IDEA с установленными плагинами:
 - su.rus.language-pack - русский язык интерфейса,
 - com.intellij.grazie.pro - проверка грамматики и орфографии,
 - org.asciidoctor.intellij.asciidoc - рабочая среда для верстки AsciiDoc,
 - org.jetbrains.plugins.gitlab - расширение для GitLab.
- Браузер [Firefox](#);
- Контроль размеров окна [Window Resizer](#);
- Снимки интерфейсов [FastStone Capture](#);
- Управление версиями GitKraken или IntelliJ IDEA.

Подготовка к работе

Каждая система которую мы документируем называется проектом.

Перед всеми манипуляциями с настройкой инструментария определитесь с местом хранения проектов на вашем компьютере. Создайте в корне любого доступного локального диска на компьютере каталог **GIT** (например, `c:\GIT\`). Там будут ваши проекты.

Регистрация на GitLab

Перейдите по адресу https://gitlab.com/users/sign_in и пройдите процедуру регистрации или авторизуйтесь если уже зарегистрированы.



Управление учетными записями на GitLab доступно только вам. Никакой администратор не поможет со входом. Если забыли учетку, логин найти можно, но пароль восстанавливать нужно самостоятельно. Не теряйте учетных данных!

Рисунок 1. Стартовая страница GitLab

GitLab.com

GitLab.com offers free unlimited (private) repositories and unlimited collaborators.

- [Explore projects on GitLab.com](#) (no login needed)
- [More information about GitLab.com](#)
- [GitLab Community Forum](#)
- [GitLab Homepage](#)

By signing up for and by signing in to this service you accept our:

- [Privacy policy](#)
- [GitLab.com Terms.](#)

First name

Last name

Username

Email

Password

Minimum length is 8 characters.

Я не робот  reCAPTCHA

Конфиденциальность - Условия использования

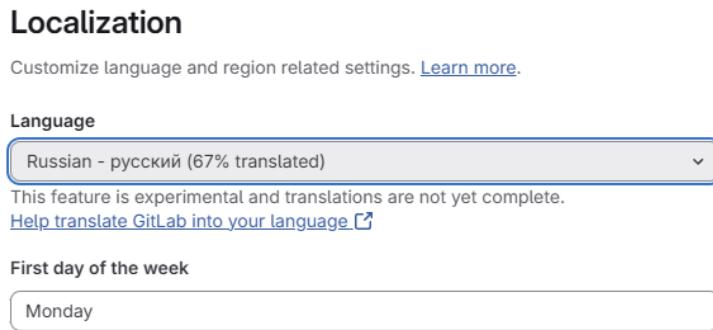
Register

By clicking Register, I agree that I have read and accepted the GitLab [Terms of Use](#) and [Privacy Policy](#)

После регистрации переключите интерфейс GitLab на русский язык.

Верхний правый угол (ваш аватар) > Preferences > User setting > Preferences > Localization

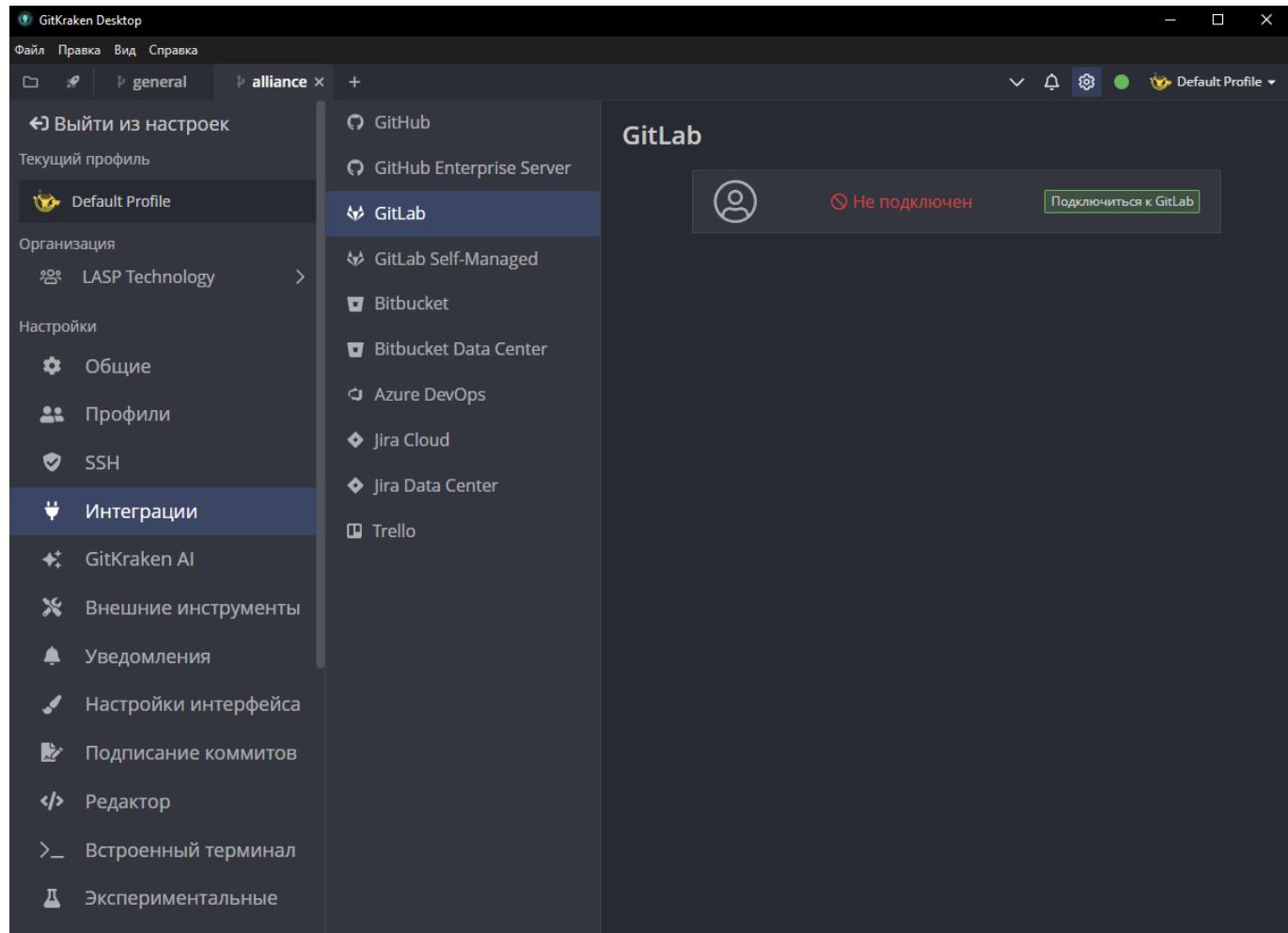
В поле language выберите **Russian**.

Рисунок 2. Меню установки языка интерфейса GitLab

Для сохранения нажмите **Save Changes** (внизу страницы) и перезагрузите страницу.

Авторизация GitKraken на GitLab

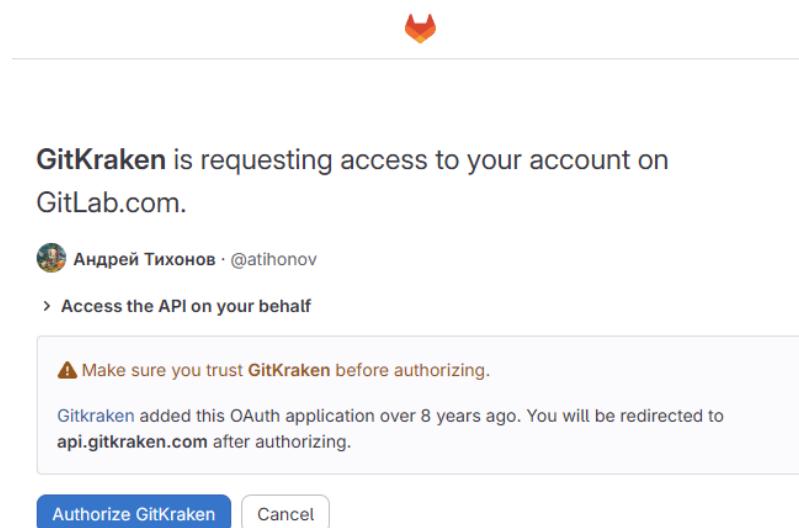
После регистрации на GitLab нужно авторизовать там GitKraken.

Рисунок 3. GitKraken до авторизации на GitLab

Для этого в GitKraken перейдите по пути **Файл > Настройки > Интеграции > GitLab** и нажмите **Подключиться к GitLab**.

В браузере будет открыта страница авторизации GitLab. Войдите со своими учетными данными.

Рисунок 4. Авторизация GitKraken на GitLab



Нажмите **Authorize GitKraken**.

Рисунок 5. Подтверждение авторизации

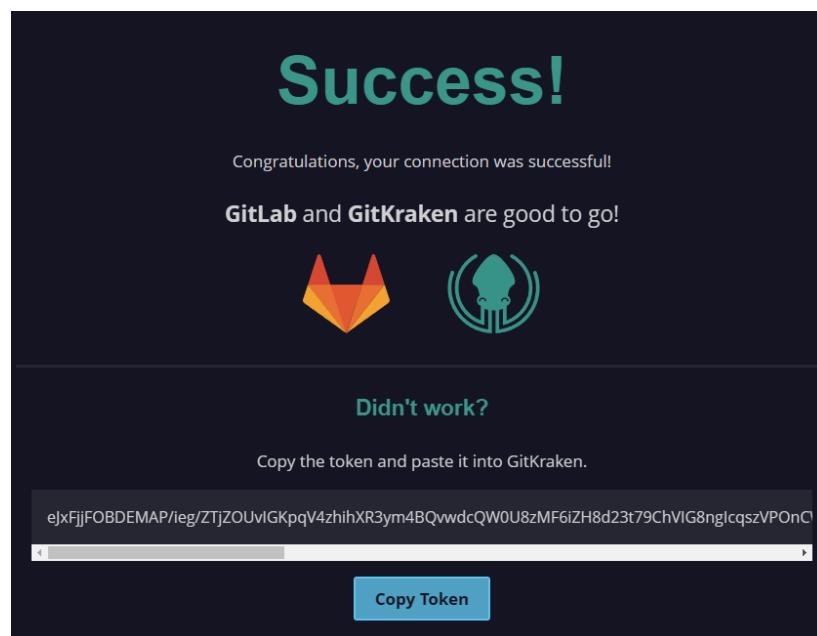
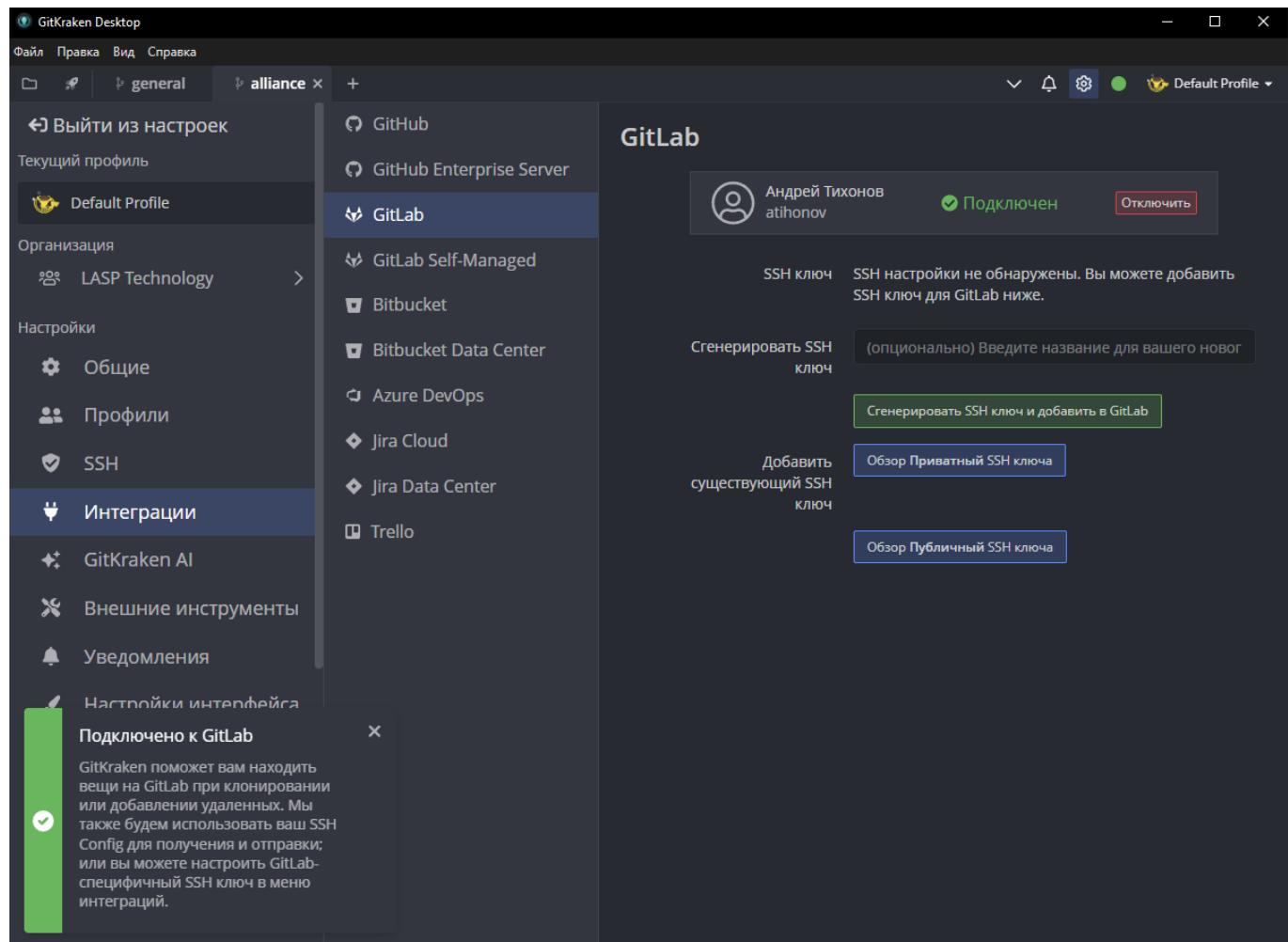


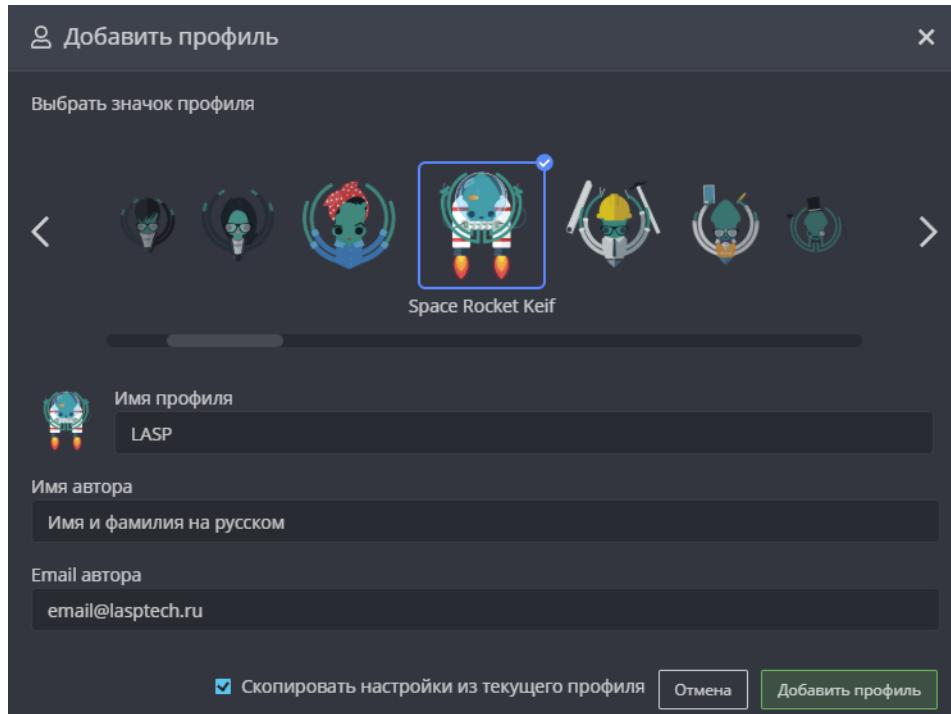
Рисунок 6. GitKraken после авторизации на GitLab



Настройка имени пользователя

Для корректного отображения вас как пользователя следует настроить профиль. Для этого перейдите по пути **Файл > Настройки > Профили > + Добавить профиль**

Рисунок 7. Профиль



По желанию выберите иконку и впишите имя профиля и заполните поля (**Имя автора** имя и фамилия на русском, **Email Автора** - свой рабочий почтовый ящик).

Проверьте параметры и нажмите **Добавить профиль**.

Выйдите из настроек GitKraken (**Выйти из настроек**) или просто нажмите на «шестеренку» в верхнем правом углу.

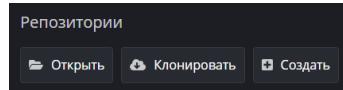
Клонирование репозитория

Каждый проект расположен в отдельном репозитории на GitLab.

Для работы с проектом нужно сделать его копию (клонировать) на вашем компьютере.

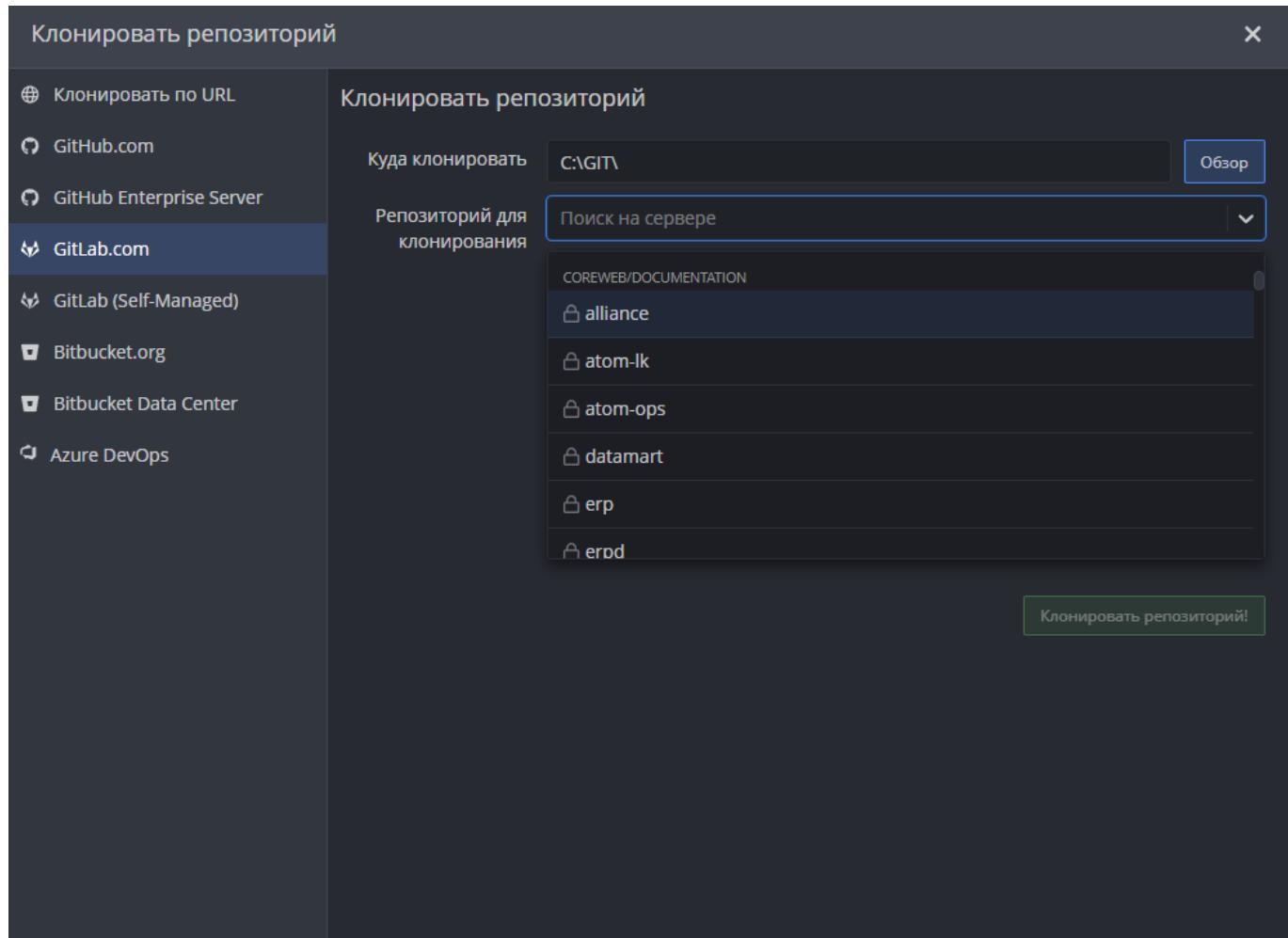
Откройте новую вкладку GitKraken **Ctrl + T** и найдите там блок управления репозиториями.

Рисунок 8. Управление репозиториями



Для клонирования необходимого проекта нажмите **Клонировать**.

Рисунок 9. Выбор проекта для клонирования



В меню открывшегося окна выберите **GitLab.com**.

В поле **Куда клонировать** укажите путь к каталогу для хранения проектов (кнопка **Browse** вызывает Проводник).

В выпадающем списке поля **Репозиторий для клонирования** найдите необходимый проект.

Нас интересуют проекты расположенные в **COREWEB/DOCUMENTATION**. Там вся документация.

Когда проект будет выбран, в поле **Полный путь** будет отображен путь сохранения.

У вас на диске должна быть строгая иерархия каталогов проектов во избежание дальнейшей путаницы.

Есть автособираемые проекты со схожей структурой, которые мы будем впоследствии использовать для включения в документы (например описания системных объектов, сервисов и тому подобного).

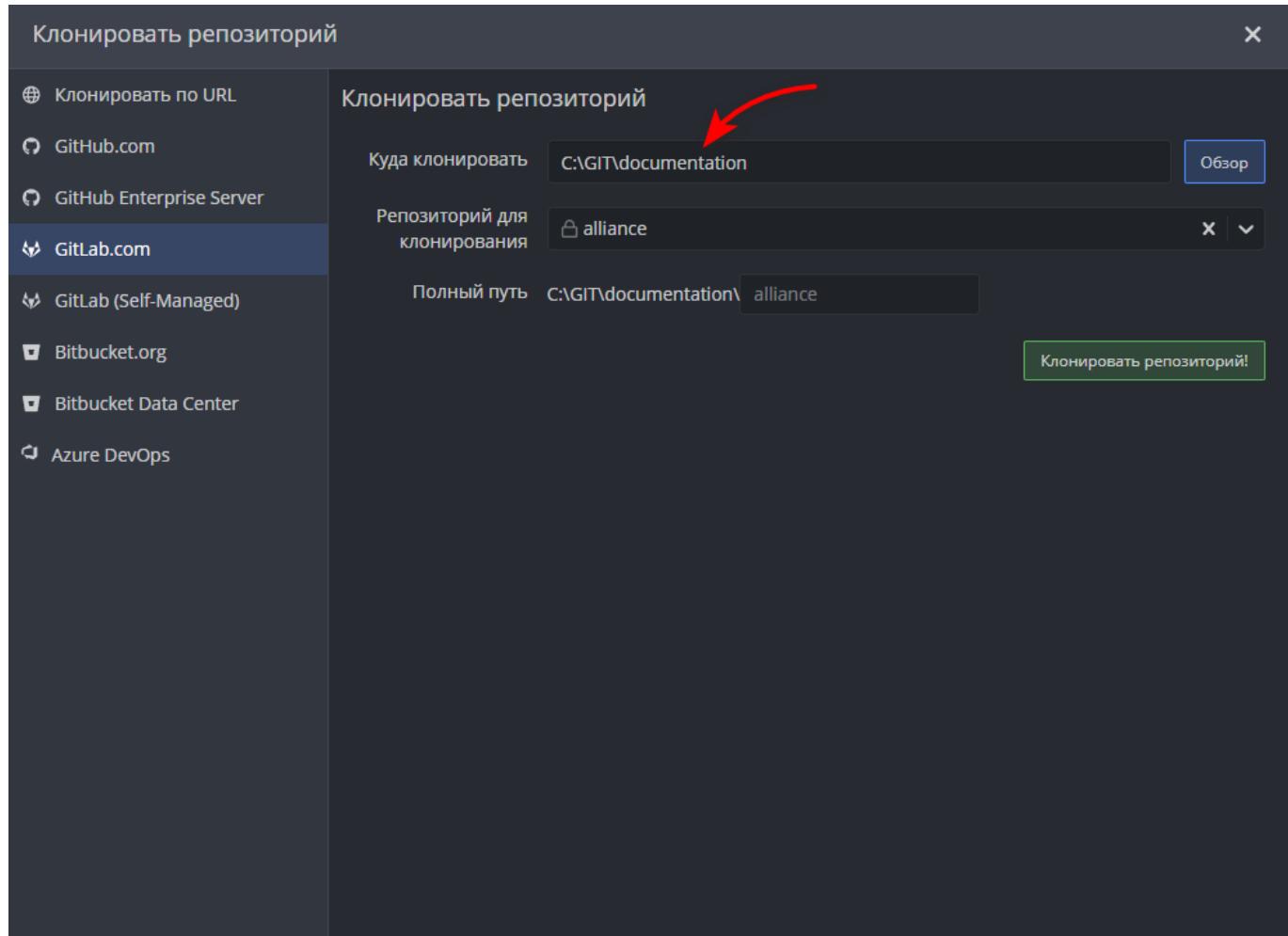
Вся документация должна храниться в каталоге документация (documentation).

Поэтому добавьте в путь **documentation** и он будет автоматически создан в процессе клонирования.

Путь в поле Полный путь

C:\GIT\documentation\<имя проекта>

Рисунок 10. Путь клонирования проекта



Проверьте пути и для начала процесса нажмите **Клонировать репозиторий!**

Дождитесь завершения клонирования и в появившемся диалоговом окне **Репозиторий <Имя проекта> успешно клонирован** нажмите **Открыть сейчас** для открытия проекта.

Также необходимо клонировать проекты **general** ([COREWEB/DOCUMENTATION/COMMON](#)) и **admin** ([COREWEB/DOCUMENTATION/ADMIN](#)).



Там находятся общие части (partials) и изображения, используемые в документах.

Если проекты **general** и **admin** не будут размещены локально, содержимое их partials не будет отображено в документах.

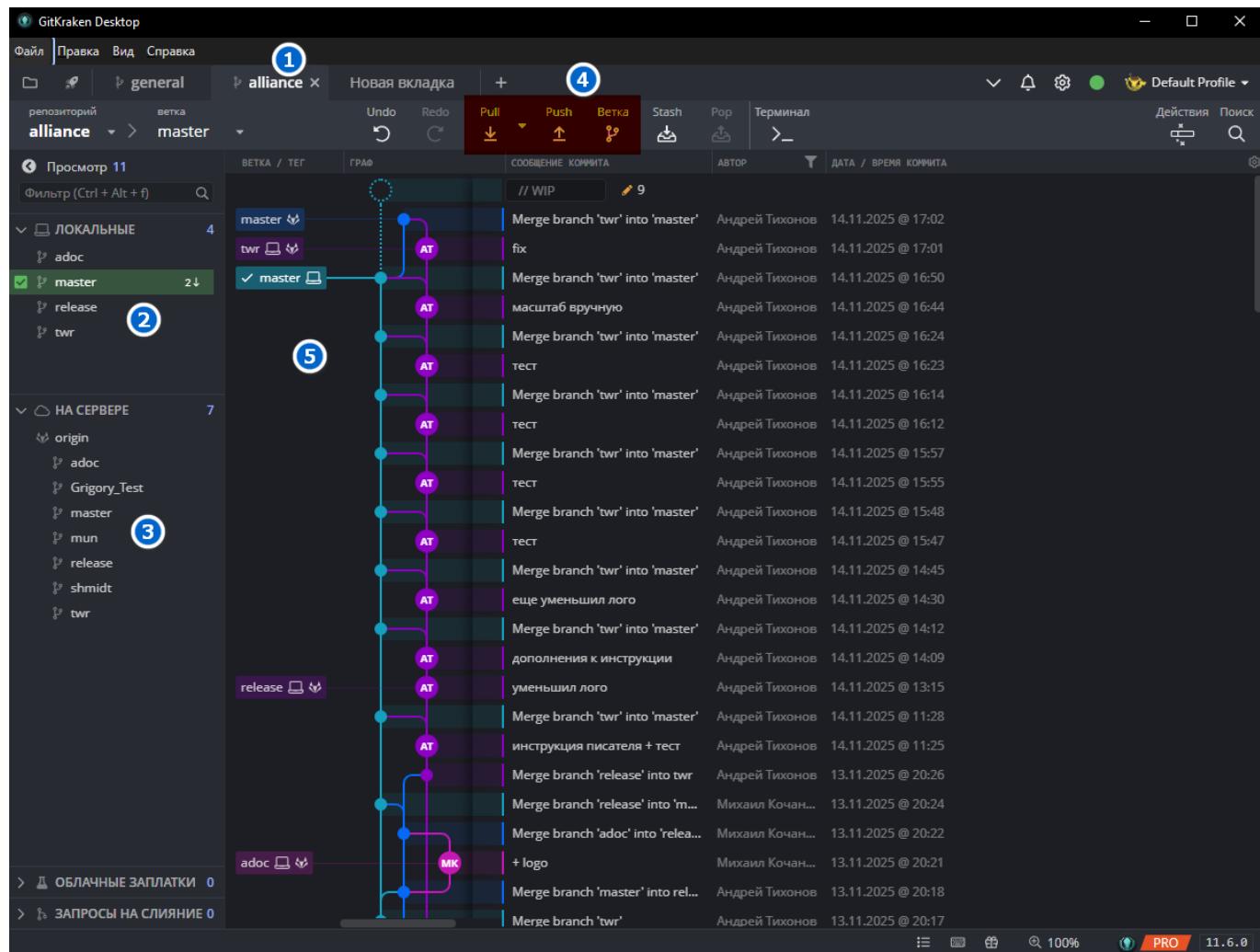
В пути следует указать **common**.

Путь в поле Полный путь

```
C:\GIT\common\admin  
C:\GIT\common\general
```

Управление версиями

Рисунок 11. Проект в GitKraken



1. Закладка проекта.
2. Локальные копии веток (на вашем компьютере).
3. Удаленные копии веток (на сервере GitLab).
4. Необходимые вам кнопки:
 - **Pull** - забрать обновленную версию с сервера (обновить);
 - **Push** - отправить локальную версию на сервер (сохранить);
 - **Ветка** - создать новую ветку в выбранном месте.
5. История изменений проекта.



Кнопка **Stash** прячет от Git изменения, а **Pop** достает спрятанное.

Начало работы

Перед началом работы необходимо:

- *нет своей ветки* - создать новую ветку от release (**Ветка** стоя на release)
- *есть своя ветка* - обновить ее (**Pull** стоя на своей рабочей ветке).

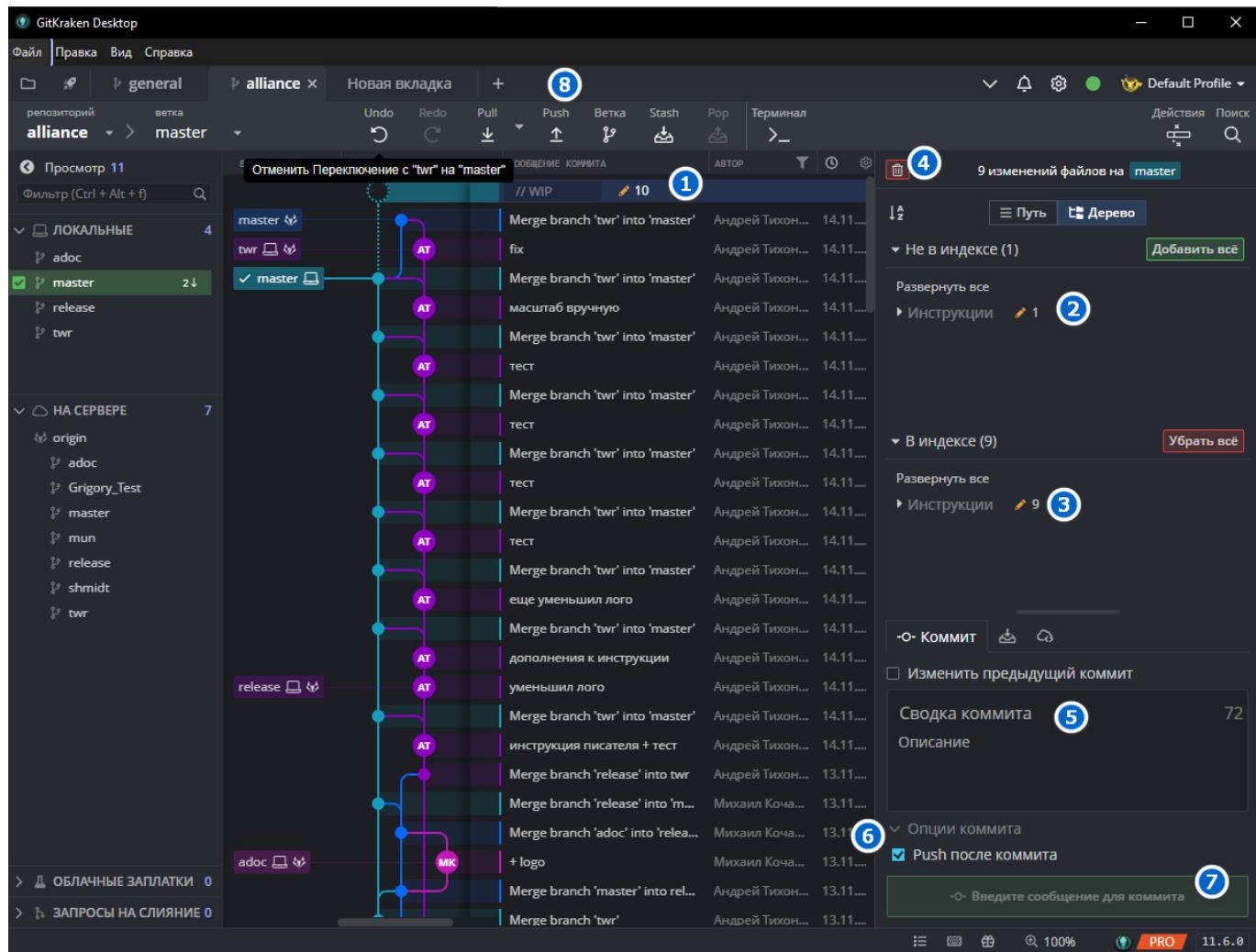
Теперь можно свернуть или даже закрыть GitKraken и заниматься своей задачей.

Фиксация изменений

Для фиксации результатов вашей работы откройте GitKraken.

Изменения фиксируются в Git в результате выполнения коммита.

Рисунок 12. Публикация изменений



Когда вы что-то изменили в проекте (создали или отредактировали файлы) GitKraken покажет это (1).

В блоке **Не в индексе** (2) список файлов не включенных в коммит. Нажмите **Добавить все** для их включения. В результате они попадут в блок **В индексе** (3). Для их исключения нажмите **Убрать все**.

Включать и исключать можно как все, так и одиночные файлы. Для этого нужно раскрыть дерево, навести указатель на файл и нажать соответственно **Добавить в индекс** или **Убрать из индекса**.



Для удаления изменений нажмите на корзину (4). Будьте внимательны! Все ваши изменения будут потеряны без подтверждения.

В блоке **Сводка коммита** (5) следует ввести наименование коммита, также можно ввести более детальное описание изменений.

Для отправки локальной версии на сервер сразу после коммита установите флажок **Push после коммита** (6). Как правило, отправка осуществляется после финального коммита по задаче нажатием на **Push** (8).

Для выполнения коммита нажмите **Коммит <количество файлов в индексе>**.

В результате внесенные вами изменения будут сохранены в Git.



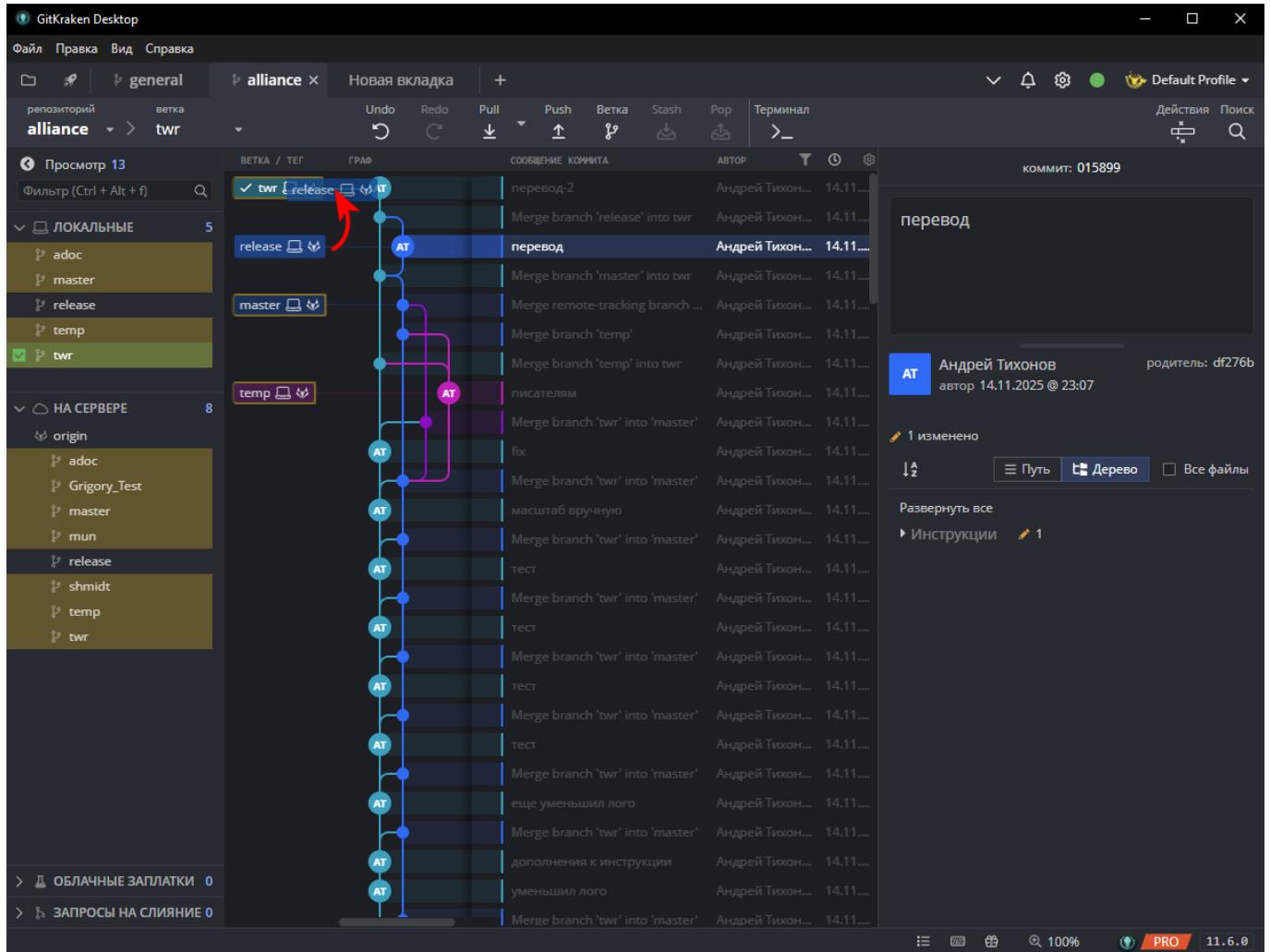
При отсутствии коммита результаты вашей работы будут потеряны!

Слияние

После того как вы выполнили финальный **Push** следует создать запрос на слияние своей ветки с веткой `master` текущего проекта.

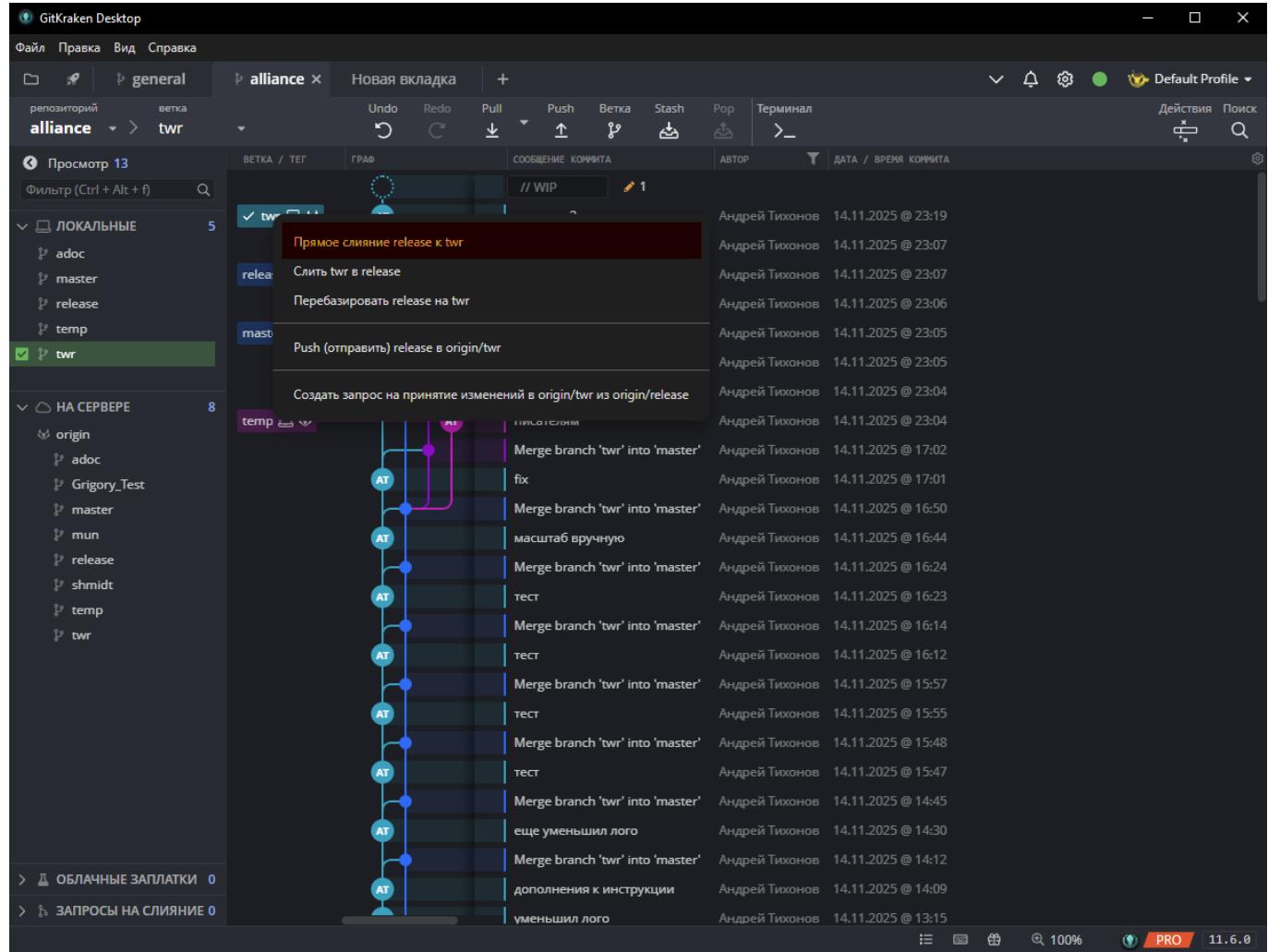
Сначала следует влить в свою ветку `release`. Для этого выделите ветку `release` и перетащите ее в свою.

Рисунок 13. Захват и перетаскивание `release` в свою ветку



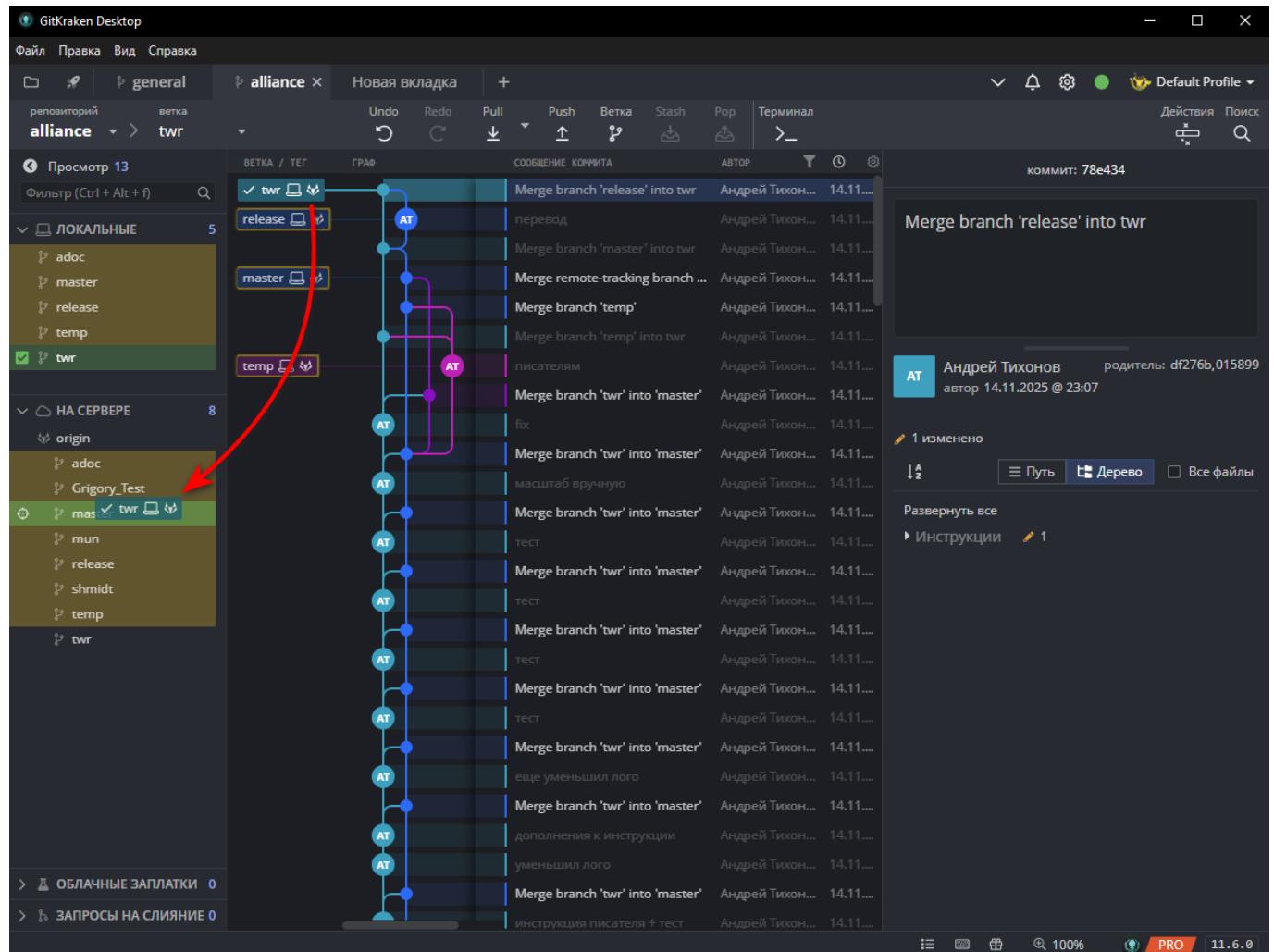
Затем в открывшемся контекстном меню выберите **Прямое слияние `release` к <имя вашей ветки>** (влить релиз в вашу ветку).

Рисунок 14. Команда слияния release в текущую ветку



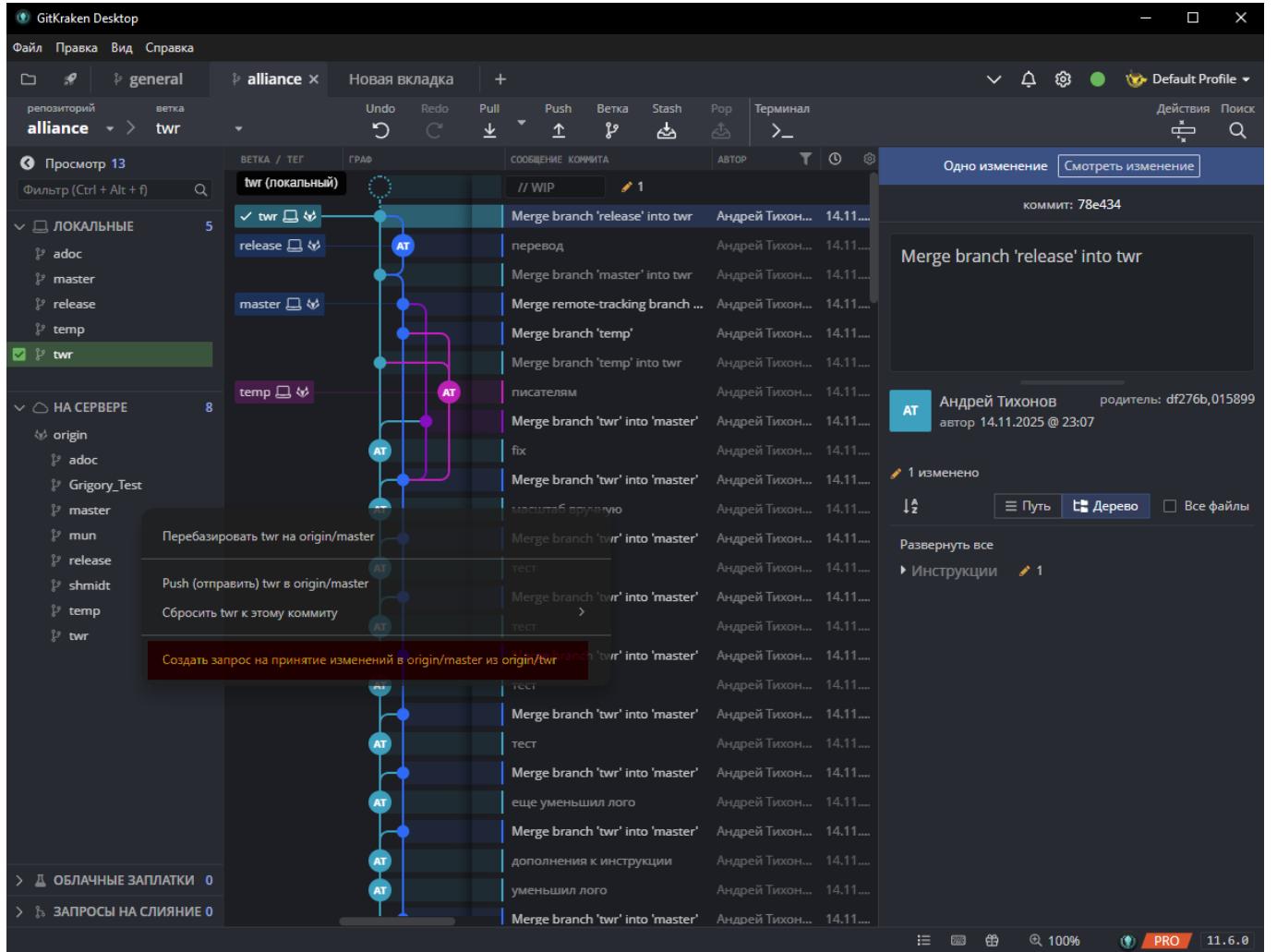
После этого выделите свою ветку и перетащите на удаленный master.

Рисунок 15. Установка направления слияния



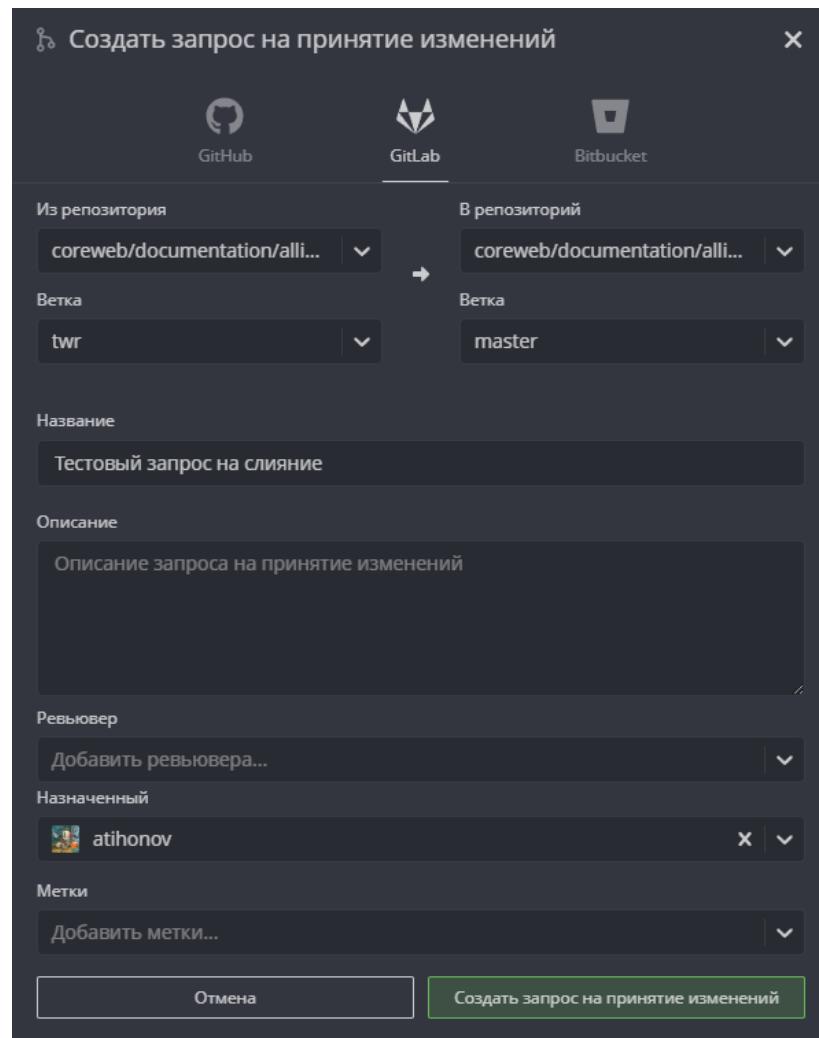
Затем в открывшемся контекстном меню выберите **Создать запрос на принятие изменений в origin/master из origin/<имя вашей ветки>** (начать слияние в мастер на сервере с вашей удаленной ветки на сервере).

Рисунок 16. Команда Создать запрос на слияние



Далее будет открыто окно параметров слияния.

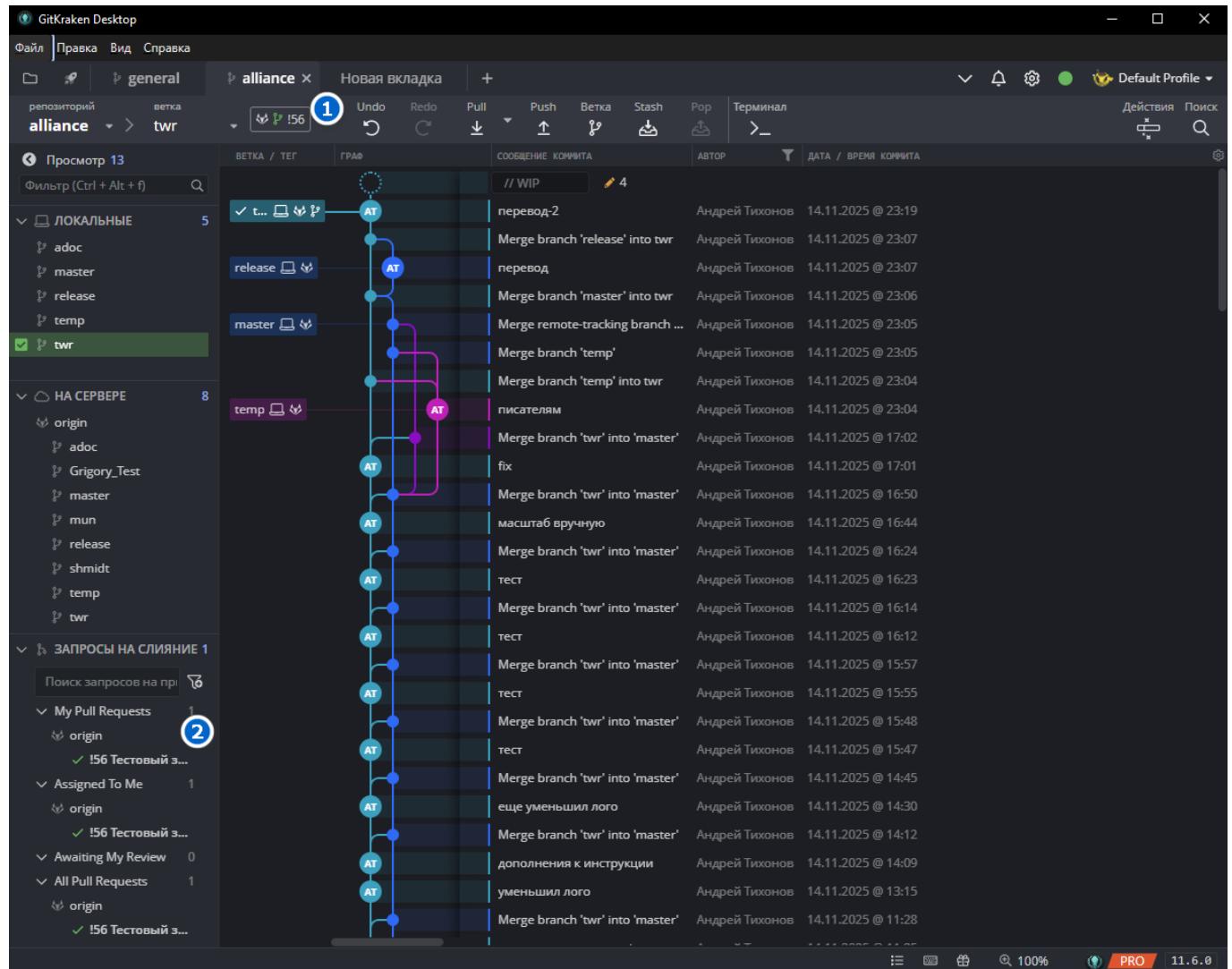
Рисунок 17. Параметры запроса на слияние



Значения (сервер, направления слияния и ветки) подставляются автоматически. Вам нужно только заполнить наименование запроса на слияние (поле **Название**). Наименование должно быть понятно принимающему слияние, которого следует выбрать в выпадающем списке **Назначенный** (принимающий).

После заполнения параметров нажмите **Создать запрос на принятие изменений**.

Рисунок 18. Запрос на слияние



Принятие запроса на слияние производится на GitLab.

Состояние вашего запроса на слияние можно увидеть на сервере. В интерфейсе GitKraken отображены ссылки на активный (не слитый) запрос на сервере (1) и (2).

Рисунок 19. Еще не принятый запрос на слияние на GitLab

Открыто Андрей Тихонов запросил(а) слияние `writertears` в `master` только что

Обзор 0 Коммиты 1 Конвойеры 2 Изменения 5

0 0

Конвойер запроса на слияние #2156563998 выполнен

Конвойер запроса на слияние выполнен для `2e2f02d9` 15 секунд назад

8✓ Отозвать одобрение Одобрено вами

Итоги тестирования: нет изменённых результатов тестирования, всего 1 тест Полный отчёт

Готово к слиянию!

Удалить исходную ветку Сжать коммиты [?](#) Изменить сообщение коммита

- Исходная ветка [на 7 коммитов позади](#) целевой ветки. [Перебазировать исходную ветку](#)
- 1 коммит и 1 коммит слияния будут добавлены к `master`.

Слияние

В результате, после проверки и принятия вашего запроса слияния, на сервере будет запущена сборочная линия по сборке итоговой документации.

В случае выявления ошибки на предварительной сборке или не принятия принимающим ввиду некачественного выполнения задачи, запрос будет в ожидании следующего коммита с внесенными правками.



В случае отсутствия работ по исправлению ошибок, запрос будет закрыт принимающим.



Новичкам можно не выполнять **Запрос на слияние**. Только коммит и **Push**, но затем обязательно нужно уведомить руководителя о завершении задачи.

Возможные конфликты

При одновременной работе над одним файлом несколькими людьми возникают конфликты при слиянии.

В случае появления конфликтов они могут быть разрешены как локально, средствами GitKraken, так и на сервере GitLab.

Это довольно хлопотно и нужно запрашивать автора изменений обо всех вхождениях при конфликте. Что отрывает участников от основной работы, тратит время и нервы.



Убедительная просьба! Не работайте над одним файлом одновременно.

Работа с изображениями

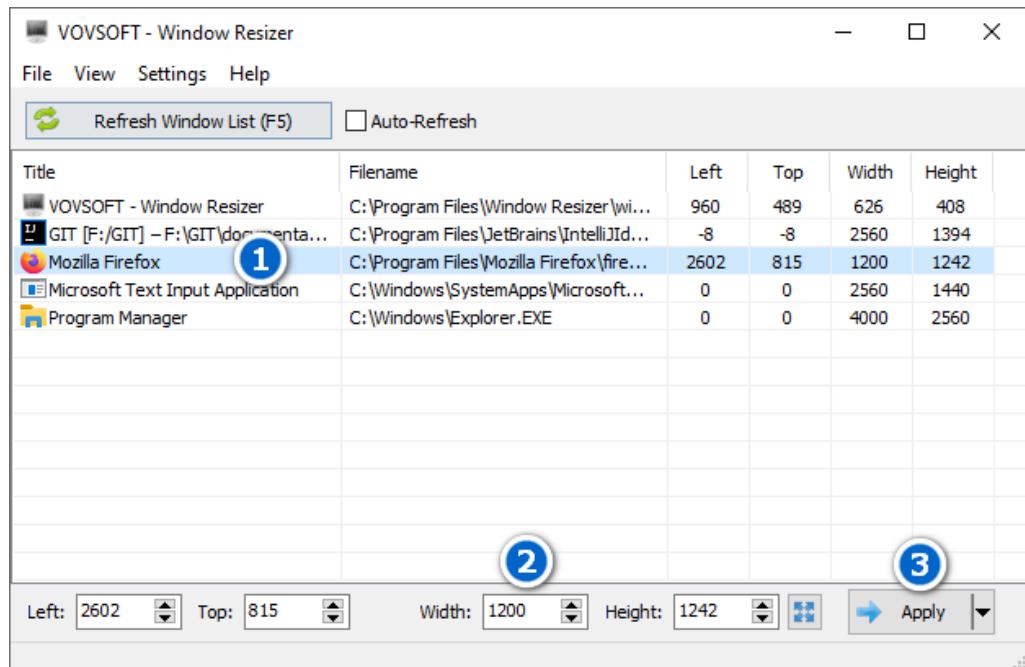
Настройка размера источника

Учитывая то, что работа над задачами выполняется разными пользователями на разных мониторах, подход к скриншотам у нас унифицировано.

Все интерфейсы должны быть сняты из браузера Mozilla Firefox в оконном виде в масштабе 100% с шириной окна 1200 пикселей.

Для контроля размера окна будем пользоваться программой Window Resizer.

Рисунок 20. Window Resizer



При запущенном браузере откройте программу, выберите в ней Mozilla Firefox (1), установите значение **1200** в поле **Width** (ширина) (2) и нажмите **Apply** (применить) (3).

В результате ширина окна браузера будет установлена.



Не забывайте перед началом работы проверять размер и масштаб браузера.

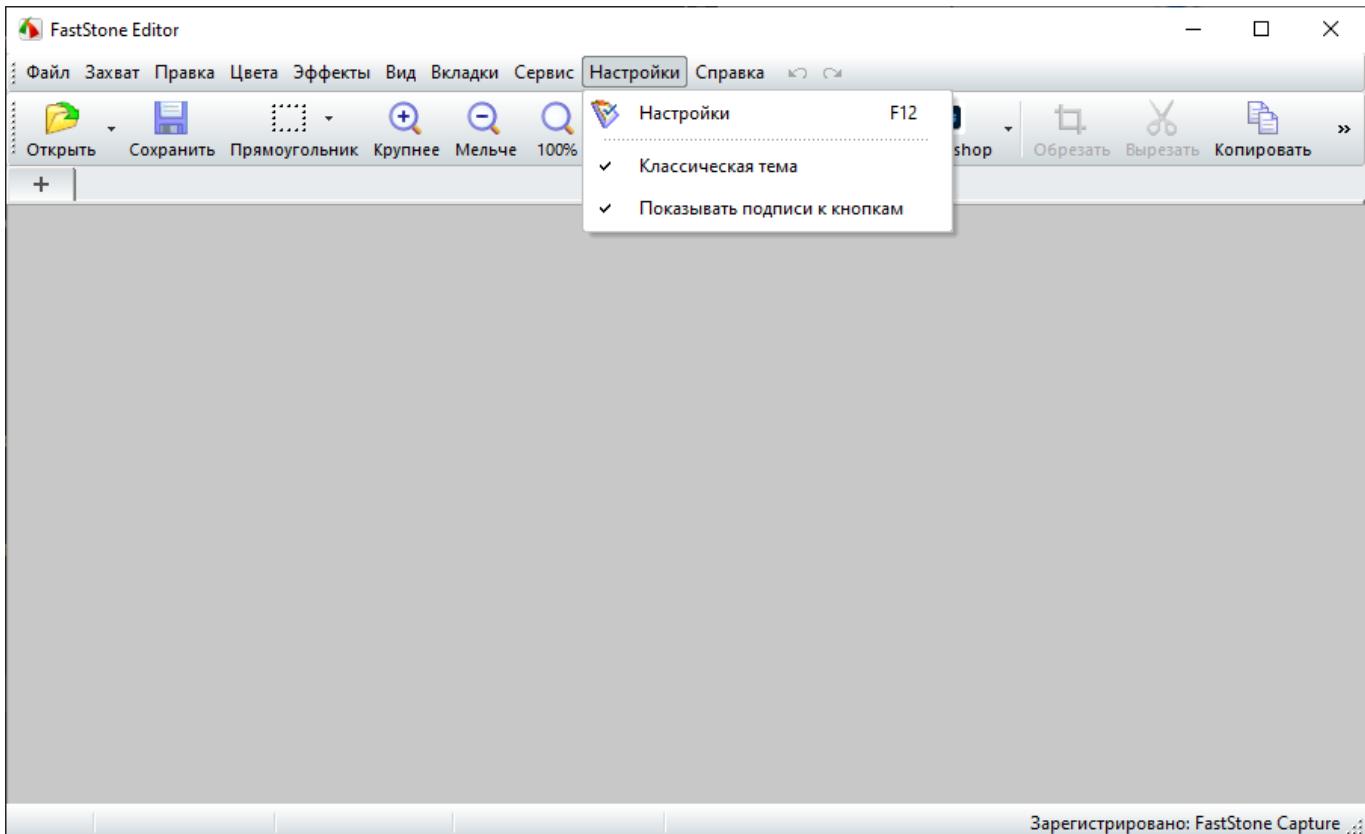
Снятие скриншота

Не нужно пользоваться Ножницами или прочими стандартными инструментами Windows типа **PrtScn** (Print screen).

Снимать интерфейсы будем с помощью программы FastStone Capture в формате .png.

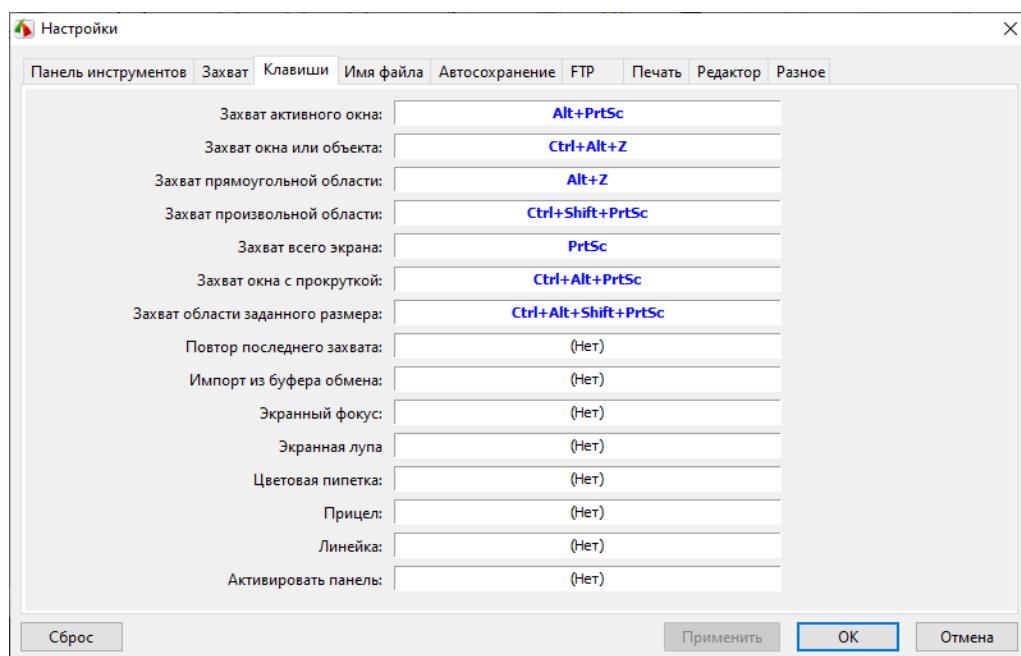
Сначала необходимо настроить «горячую клавишу» для снятия области экрана. Для этого перейдите в настройки.

Рисунок 21. Переход в настройки FastStone Capture



Установите удобное для вас сочетание клавиш в поле **Захват прямоугольной области**. У меня **Alt + Z** (всегда под левой рукой).

Рисунок 22. Настройка горячих клавиш FastStone Capture



Остальные настройки можно не трогать.

Теперь двумя кнопками легко можно снимать любую часть интерфейса.

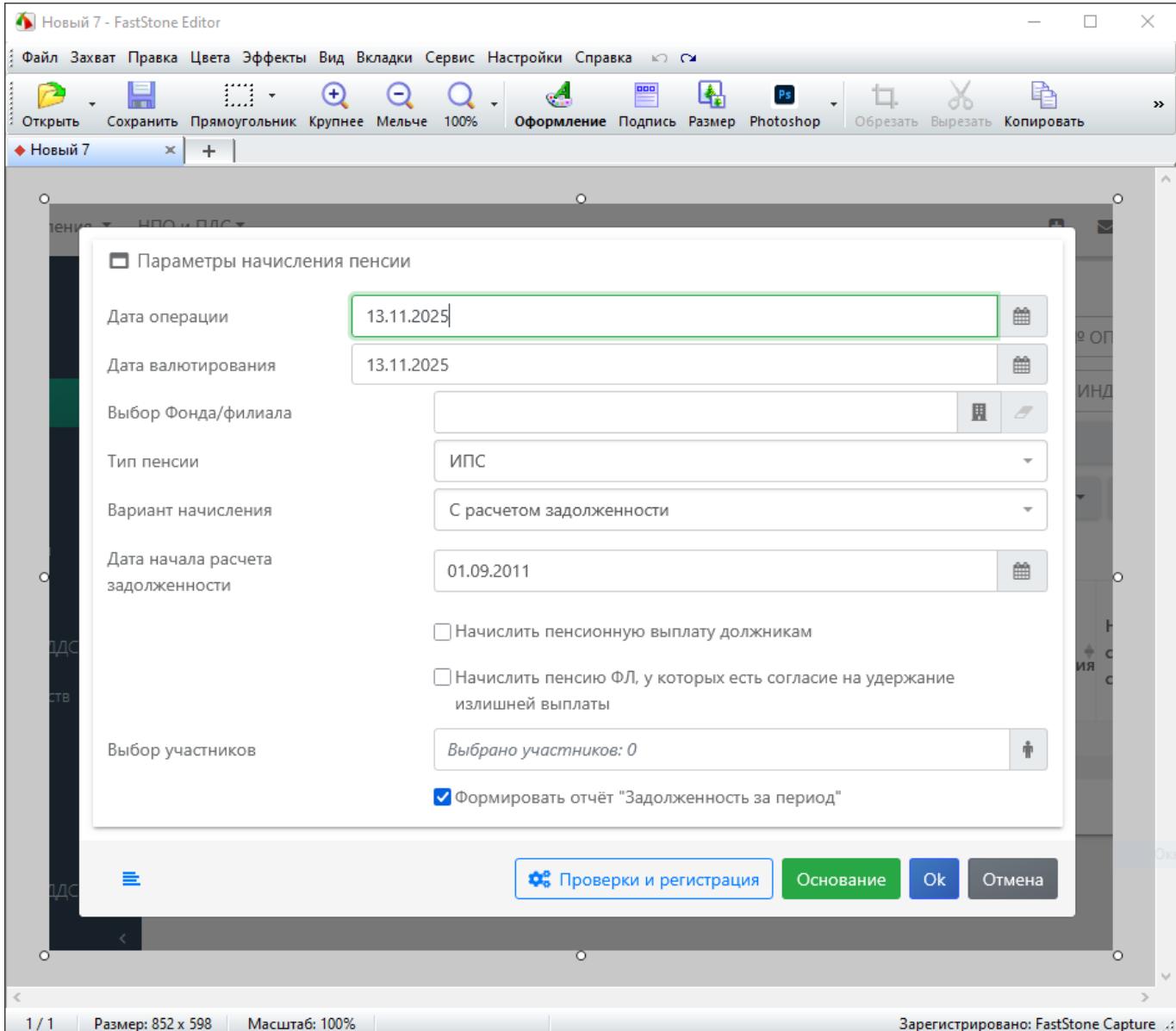
После снятия области экрана изображение попадает в окно редактора FastStone Capture.

Оттуда вставим его в документ. Но об этом позже.



Можно захватить большую область и затем подогнать границы рамками (кадрировать) в FastStone Capture.

Рисунок 23. Кадрирование изображения



Правила снятия интерфейсов:



- снимайте с открытым боковым меню, но без его захвата;
- оставляйте серую рамку размером 1-2 пикселя (на глаз, но аккуратно);
- в таблицах указывайте отображение 5 строк;
- для отображения всех доступных кнопок выделяйте первую запись в списке.

Рисунок 24. Интерфейс



Модальные окна снимайте без рамки, но аккуратно (полукруглые углы не должны попадать в кадр).

Рисунок 25. Модальное окно



Размытые кнопки, снимайте с внешней рамкой 1-2 пикселя. На белом в документе будет смотреться без резких переходов.

Правила размещения изображений в каталогах будут приведены в разделе **Порядок работы. Включение изображений**.

Основной редактор

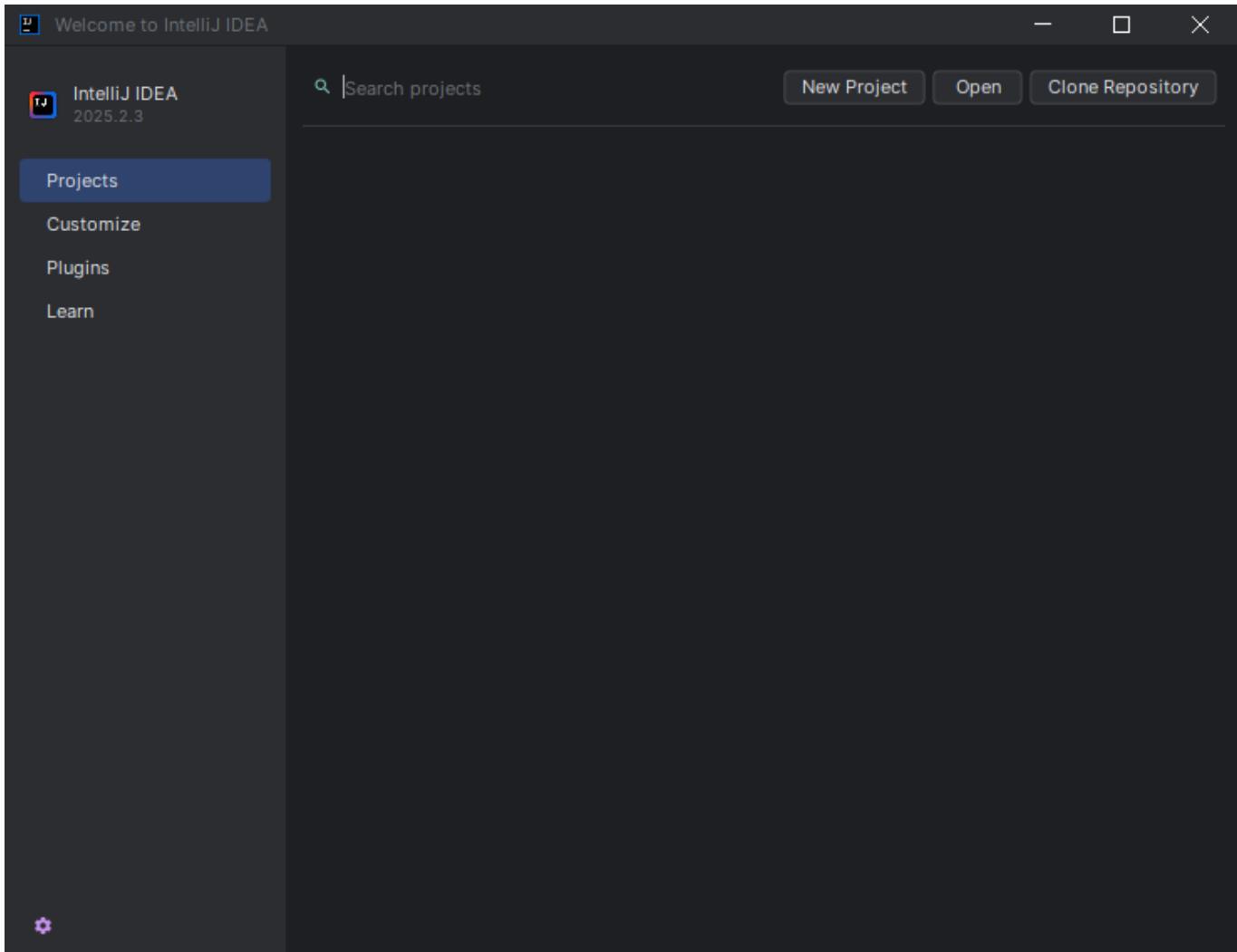
Работать будем в IntelliJ IDEA. Он нам наиболее подходит. Лучше инструментов нет.

Каталог как проект IntelliJ IDEA

Для того чтобы редактор следил и учитывал изменения по всем нашим проектам следует создать единое пространство для их размещения.

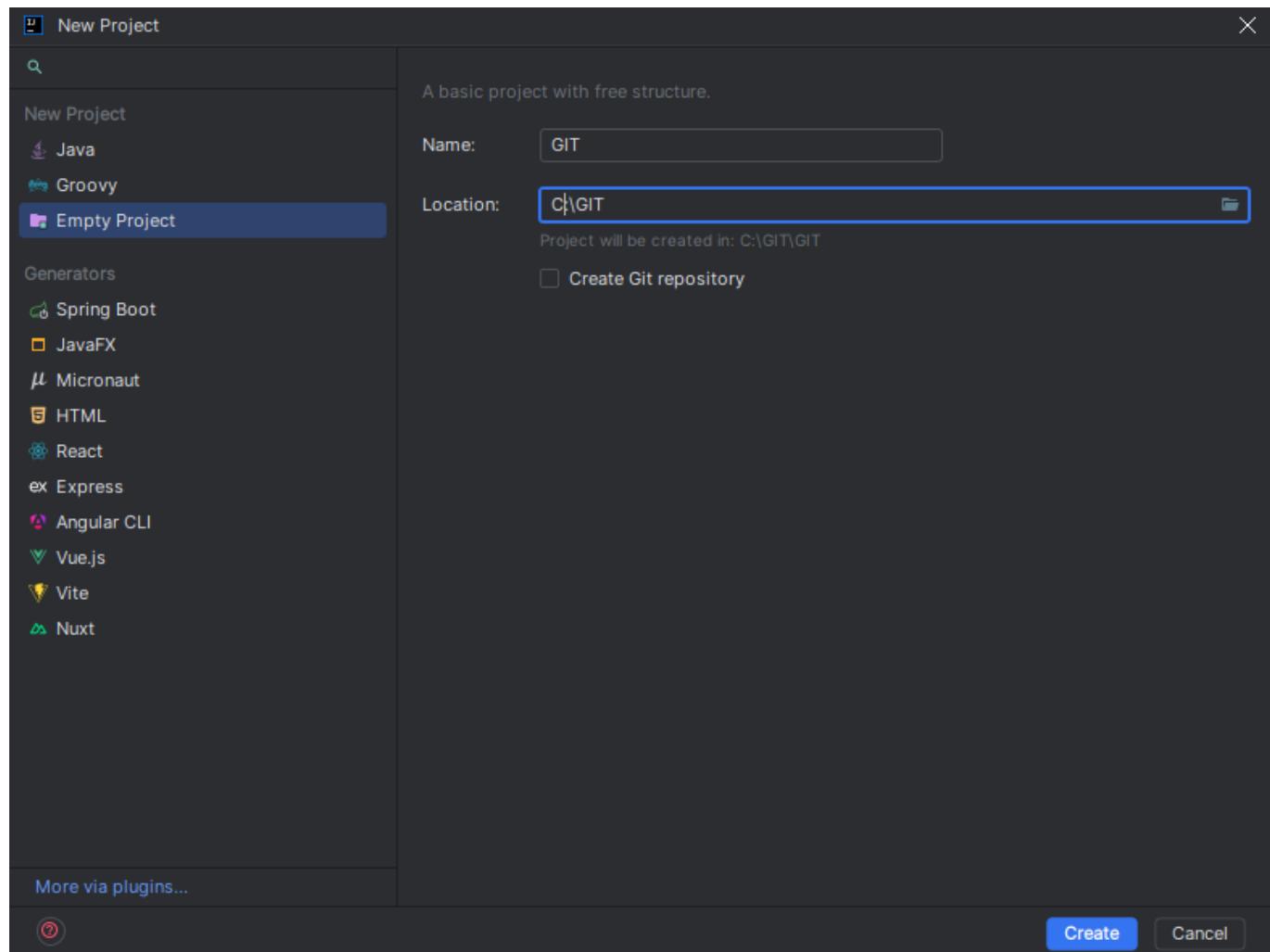
Откройте IntelliJ IDEA.

Рисунок 26. Приветственная страница IntelliJ IDEA



Создайте пространство для работы со всеми нашими проектами **Project > New Project**.

Рисунок 27. Параметры проекта



1. Выберите **Empty Project** (Пустой проект);
2. Заполните поле **Name** (Имя проекта в IntelliJ IDEA);
3. Укажите расположение созданного каталога GIT на своем диске.
4. Нажмите **Create** (Создать).

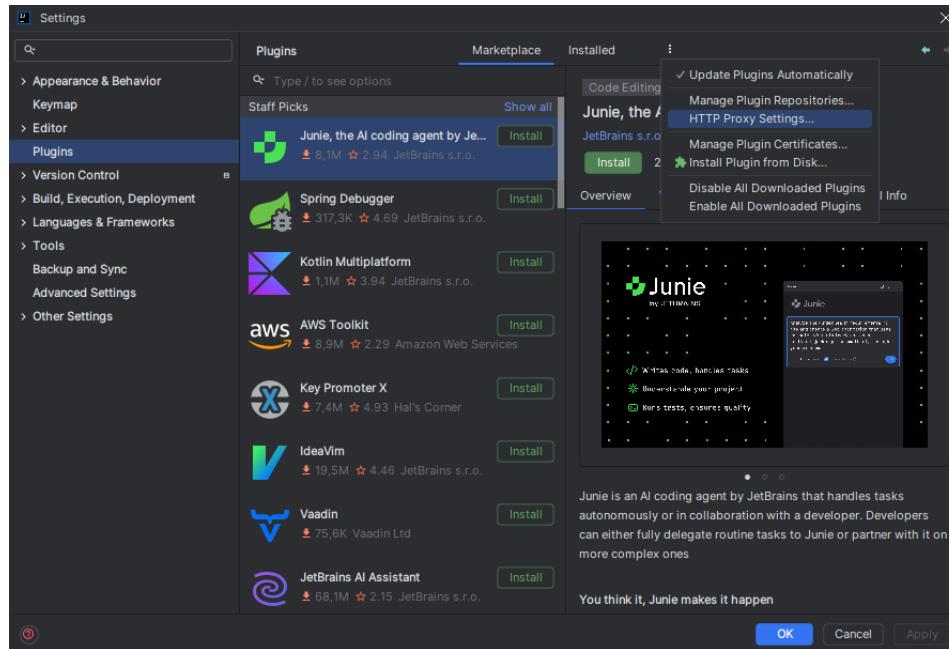
В результате пространство будет создано и ранее клонированные в GitKraken проекты будут отображены в виде дерева каталогов в левой части интерфейса.

Начальные настройки

В начале необходимо установить необходимые плагины в редактор.

File > Settings > Plugins

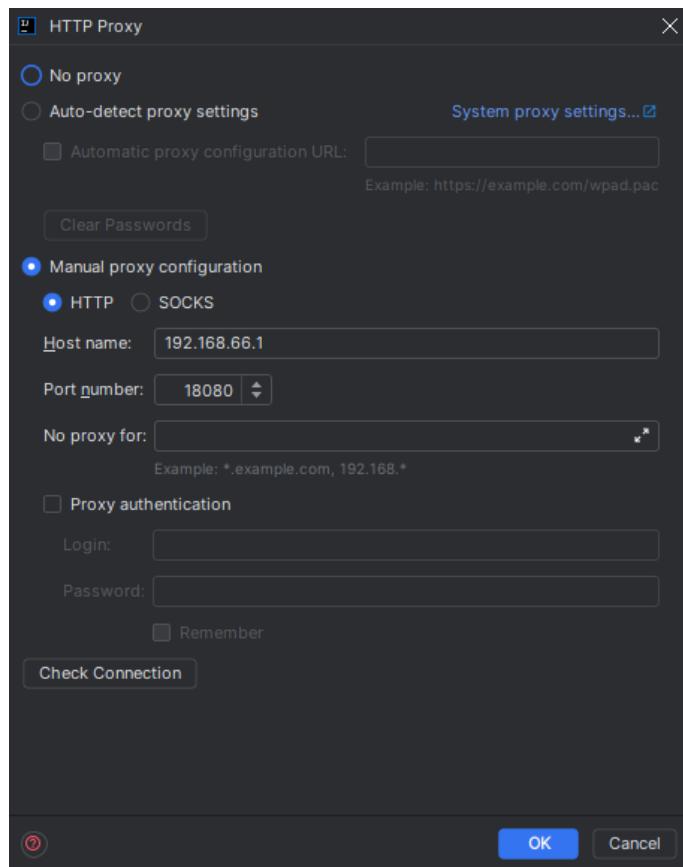
Рисунок 28. Плагины



В меню выберите **HTTP Proxy Settings** и в открывшемся окне введите значения:

- Host name - 192.168.66.1
- Port number - 18080

Рисунок 29. Окно HTTP Proxy

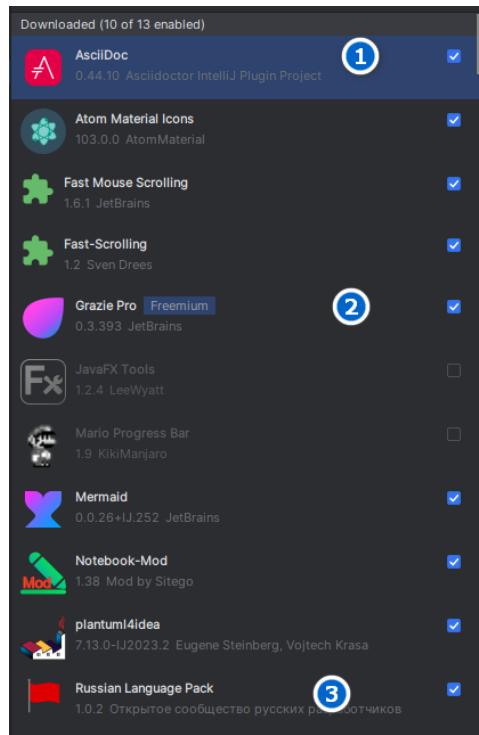


Далее необходимо найти и скачать следующие плагины:

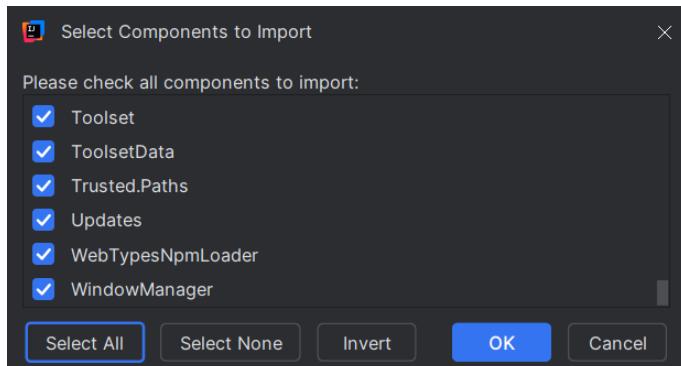
- AsciiDoc - рабочая среда для верстки AsciiDoc,
- Grazie Pro - проверка грамматики и орфографии,
- Russian Language Pack - русский язык интерфейса.



Для скачивания обновлений и плагинов должно быть активным подключение к vpn.lasptech.ru.

Рисунок 30. Окно с установленными плагинами

Затем импортируйте предоставленную конфигурацию. **File > Manage IDE Settings > Import Settings** и выберите файл конфигурации.

Рисунок 31. Выбор импортируемых настроек

Нажмите **OK**, подтвердите перезагрузку редактора и дождитесь его запуска.

После импорта конфигурации переключите интерфейс на русский язык.

Для этого перейдите **File > Settings > Appearance & Behavior > System Settings > Language and Region**. Выберите **Russian** и подтвердите перезагрузку редактора.

Для отображения стилей поместите файлы `.asciidoctorconfig` и `site.css` в корневой каталог вашего проекта IntelliJ IDEA.

Расположение файлов стиля

```
C:\GIT\.asciidoctorconfig
C:\GIT\site.css
```

При необходимости в файле `.asciidocorconfig` измените букву диска.

Содержание файла `.asciidocorconfig`

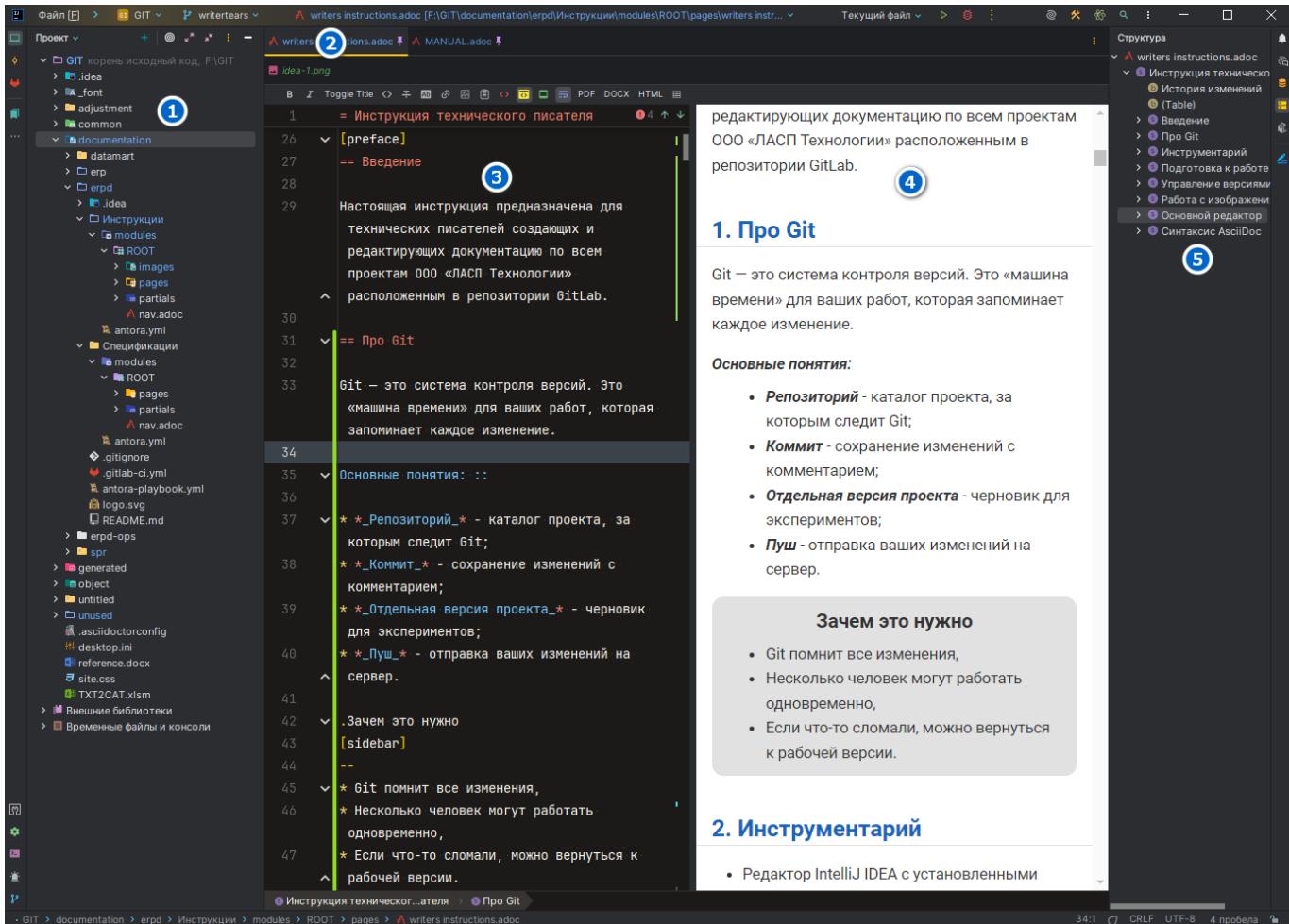
```
:stylesdir: C:/GIT
:stylesheet: site.css
:icons: font
```

Как тут все устроено

В редакторе отлично реализована функциональность горячих клавиш.

Обращайте внимание на подсказки в меню и запоминайте необходимые команды. Это позволит значительно ускорить работу.

Рисунок 32. Интерфейс редактора IntelliJ IDEA



Дерево каталогов (1):

Каждый проект состоит из корневого каталога (пример `erpd`) и включает в себя каталоги **Инструкции** и **Спецификации**, для хранения соответствующих типов документов, содержащие каталоги **modules**.

Каталоги **modules** содержат вложенные каталоги:

- `images` - для хранения изображений;
- `pages` - для хранения итоговых документов;
- `partials` - для хранения частей, включаемых в документ.



Правила наименования изображений - имена на русском, пробелы заменяются на дефис.

Вкладки (2):

Данная вкладка отображает активные и ранее открытые документы. Любая вкладка может быть закреплена (например текущий рабочий документ).

Область редактирования (3):

Основная рабочая область редактора. Там вы и будете редактировать текст и вставлять ссылки.

Область просмотра (рендер) (4):

В этой области отображается редактируемый документ почти в таком виде, в каком он будет опубликован, с включением частей и изображений. Место документа в области просмотра соответствует выбранному месту в области редактирования.

Структура документа (5):

Здесь отображена структура редактируемого документа. Если мешает можно скрыть (в правом верхнем углу).

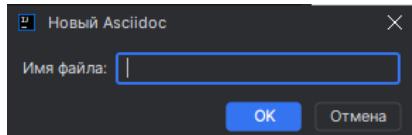
Порядок работы

Начало

К работе в редакторе можно приступать после того как вы обновили проект general (ветка release в нем синхронизирована) и создали в текущем проекте новую ветку (от release) или обновили свою ветку (**Pull**) в GitKraken.

Создайте в редакторе файл **Наименование операции.adoc** в каталоге **pages** текущего проекта.

Для этого нужно выделить каталог **pages** и выбрать в контекстном меню **Создать > AsciiDoc**.



Впишите наименование файла и нажмите **OK**.

В результате будет создан файл с необходимыми атрибутами.

Далее для доступности вашего документа на главной странице пропишите его в **index.adoc**, расположенном в каталоге **pages**:

```
* xref:Наименование операции.adoc[Наименование операции]
link:_exports/наименование-операции.pdf[.btn-pdf]#btn:[pdf]#
```



В ссылке на будущий pdf файл допускаются символы только в нижнем регистре.

Пробел должен быть заменен на дефис.

Для будущего создания pdf из вашего файла пропишите в **nav.adoc**, расположенный уровнем выше:

```
* xref:Наименование операции.adoc[Наименование операции]
```

После этого можно начинать редактирование созданного файла.

Структура операционной инструкции

Операционная инструкция предназначена для пользователей-операционистов и должна без технических деталей описывать пошаговое прохождение операции.

Инструкция должна содержать вводную часть и следующие аспекты:

1. Для чего это нужно (назначение и цель операции - «зачем мы это делаем»);
2. Где это делать (место в системе - «как найти, где нажимать кнопки»);
3. Как это сделать (пошаговые действия - «какие кнопки нажимать»);
4. Что получится (проверка результата - «убедиться, что все правильно сделал»).

Таблица 1. Структура инструкции

№	Раздел	Описание
1	Введение	Ссылка (всегда единая подключаемая часть из partials текущего проекта): <code>include::partial\$Введение.adoc[]</code>
1.1	Компетенция оператора	Ссылка (всегда единая подключаемая часть из partials general): <code>include::general::partial\$компетенция оператора.adoc[]</code>
1.2	Условия выполнения	Ссылка (всегда единая подключаемая часть из partials general): <code>include::general::partial\$условия выполнения операции.adoc[]</code>
2	Назначение	Описание основной причины для выполнения операции (составляется на основании бизнес-требований)
3	Порядок выполнения	Путь к интерфейсу входа в операцию (пишем руками путь в системе до кнопки запуска операции).
3.1	Создание операции	Кнопка запуска и вызов окна операции (пишем руками). Если планировщик, добавляем ссылку (всегда единая подключаемая часть из partials general): <code>include::general::partial\$планировщик кратко- 1.adoc[].</code>
3.1.1	Документ-основание	Ссылка (всегда единая подключаемая часть из partials general): <code>include::general::partial\$документ-основание- кратко.adoc[]</code>
3.1.2	Параметры операции или Загрузка файла	Все подробности о заполнении полей, внутренних модальных окнах и полей там (пишем руками).

№	Раздел	Описание
3.1.3	Редактирование, Отмена (опционально если есть)	Если есть возможность редактирования или отмены заполняем отдельными пунктами на этом уровне.
3.1.4	Контроль корректности данных (опционально в случае выполнения тестов данных на форме)	Ссылки (всегда единые подключаемые часть из partials general): <code>include::general::partial\$проверка корректности тест.adoc[]</code> <code>include::general::partial\$выбор пакета проверок.adoc[]</code> <code>include::general::partial\$просмотр проверок тест.adoc[]</code> <code>include::general::partial\$перечень проверок тест ссылка.adoc[]</code>
3.2	Выбор способа обработки (опционально если есть)	Ссылка (всегда единая подключаемая часть из partials general): <code>include::general::partial\$выбор способа обработки.adoc[]</code>
3.3	Аудит (опционально если есть проверки в планировщике)	Ссылки (всегда единые подключаемые часть из partials general): <code>include::general::partial\$проверка корректности.adoc[]</code> <code>include::general::partial\$просмотр проверок.adoc[]</code> (если доступно) <code>include::general::partial\$перечень проверок ссылка.adoc[]</code>
3.4	Регистрация задания если планировщик. Иначе Регистрация изменений	Ссылка (всегда единая подключаемая часть из partials general): <code>include::general::partial\$регистрация задания.adoc[]</code> или <code>include::general::partial\$регистрация изменений.adoc[]</code> или <code>include::general::partial\$регистрация изменений тест.adoc[]</code> (если тест на регистрации).
3.4.1	Управление заданием (опционально если планировщик)	Ссылка (всегда единая подключаемая часть из partials general): <code>include::general::partial\$планировщик кратко-1.adoc[]</code>

№	Раздел	Описание
3.4.2	Просмотр результата обработки (опционально если планировщик)	Ссылки (всегда единые подключаемые часть из partials general): <code>include::general::partial\$журнал 1.adoc[]</code> <code>include::general::partial\$журнал 2.adoc[]</code> <code>include::general::partial\$журнал 3.adoc[]</code> (опционально если есть отчеты)
4	Результат выполнения	Берем информацию с расследованием из ТЗ, старых инструкций и от аналитика (всегда руками)
4.3	Просмотр результата в мониторе	Ссылка (всегда единая подключаемая часть из partials general): <code>include::general::partial\$просмотр результата.adoc[]</code>
5	Приложение	Просто заголовок
5.1	Описание пакетов проверок	Таблицы с проверками теста или аудита (собираем пока вручную из ТЗ, старых инструкций и от аналитика с расследованием).
5.2	Структура файла (опционально если требуется)	Структура файла текстовым слоем в кодовом блоке.
5.3	Вид печатной формы (опционально если требуется)	Снимаем только pdf с учетом размещения на А4 (в интерфейсе EXCEL данные не читаемы).

Редактирование текста

Файл должен редактироваться в разметке AsciiDoc (возможности приведены в разделе **Синтаксис AsciiDoc**).

Для форматирования выделенного текста используются стандартные клавиши. Также можно использовать контекстное меню (AsciiDoc) или панель, расположенную в верхней части области редактирования.

Выделенный текст можно перемещать по документу перетаскиванием.

IntelliJ IDEA поддерживает подсветку новых и измененных файлов, а также файлов содержащих ошибки (орфография, ссылки). Это помогает легко найти файл по цвету подсветки начиная от корня проекта.

Есть поиск **Ctrl + F** и замена **Ctrl + R** по документу, а также по текущему проекту или всем проектам **Ctrl + Shift + F** и **Ctrl + Shift + R** соответственно.

Можно выделить и «на лету» отредактировать одинаковые вхождения в документе по **Alt + J**.
Также есть предустановленные шаблоны (кавычки, скобки, тэги), применяемые к выделенному тексту по **Ctrl + Alt + J**.

И еще там много всякого интересного.



При наличии Git как бы вы не хотели сломать проект, у вас ничего не получится.

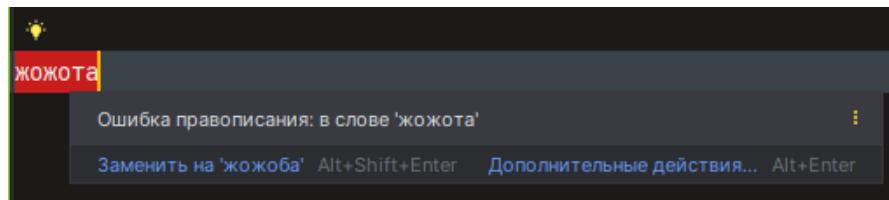
Внутри IntelliJ IDEA есть внутренний Git-клиент. Можно работать и там, но в GitKraken проще и нагляднее.

Контроль ошибок

В редакторе подключена проверка русской грамматики и орфографии посредством словаря, в который можно добавить незнакомое ему слово.

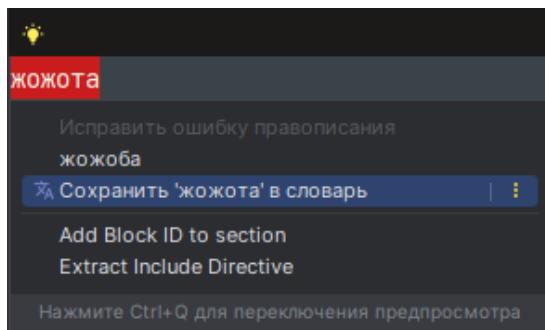
Исправление производится при наведении на ошибку из контекстного меню **Заменить на...**.

Рисунок 33. Наведение на ошибку



Добавление в словарь производится из того же меню с переходом в **Дополнительные действия** и там **Сохранить ... в словарь**.

Рисунок 34. Сохранить новое слово в словарь



Для форматирования выделенного текста используются стандартные клавиши. Также можно использовать контекстное меню (AsciiDoc) или панель, расположенную в верхней части области редактирования.

Включение partials

Для чего это нужно? Представьте что вам нужно изменить один одинаковый абзац или скриншот во многих документах. Вы просто редактируете текст в partials. Изменения будут во всех файлах содержащих эту вложенную часть.

Это намного проще и быстрее, чем открывать и править в Word 50 файлов.

Используемые во многих документах части расположены в каталоге partials текущего проекта или проекта **general** и являются своеобразной библиотекой.

В документ такие части вставляются ссылками. Структура ссылок едина и приведена в разделе **Синтаксис AsciiDoc. Части (partials)**.

Как правило, ссылки, на partials использованные ранее в других документах, просто копируются оттуда в новый.

Пример ссылки на partials внутри проекта

```
`include::general::partial$взаимодействие со смежными системами.adoc[]`
```

Пример ссылки на partials в проекте general

```
`include::general::partial$регистрация задания.adoc[]`
```

Пример ссылки на partials в проекте admin

```
`include::admin::partial$Профиль.adoc[]`
```

Пример ссылки на partials в объектах (используется в спецификациях)

```
`include::generated:objects:partial$contract/BATCH.INPAYMENT_LOADER_FORM_207_FL_DDS/AUDIT.I  
NPAYMENT_LOADER_FORM_207_FL_DDS/PACKAGE.NRD_PACKAGE.checks.adoc[]`
```



Апострофы " ` " в ссылках ставить не нужно. Здесь они даны для запрета рендера и отображения структуры ссылки в финальном документе.

Создание partials

В случае повтора в двух документах одинакового текста, нужно самостоятельно создавать partials по примеру расположенных там файлов.

На новые partials путь в документе необходимо указывать самостоятельно.



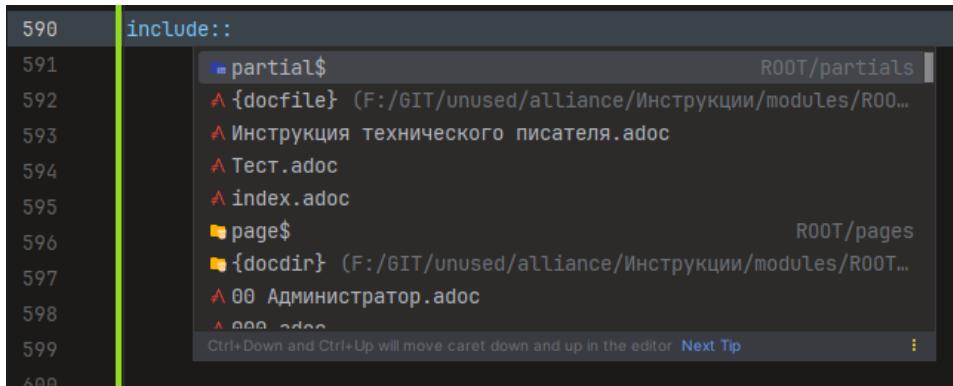
В partials должны находиться только части, входящие в текущий проект.

В **general** и **admin** расположены общие части для всех проектов.

Для вставки partials введите **include::** в необходимом месте документа. Будет открыт

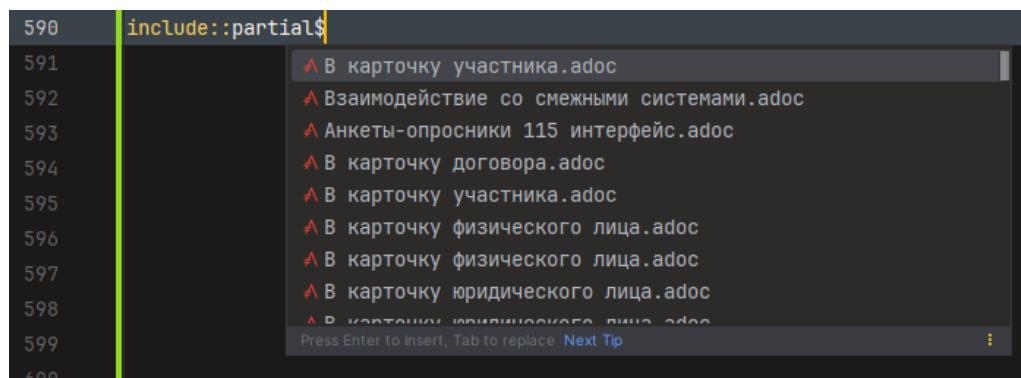
выпадающий список для выбора пути к файлу.

Рисунок 35. Выбор начала пути



Выберите значение (как правило **partial\$**). В выпадающем меню выберите необходимый partials для вложения.

Рисунок 36. Выбор файла



Закройте строку квадратными скобками **[]**

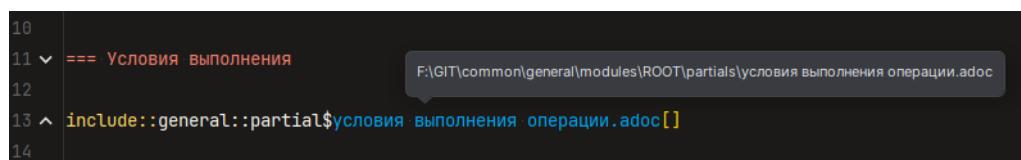
Текст вставленной части тут же отображается в области просмотра и вы сможете видеть полный текст документа.

Контроль ссылок

При ошибочном подключении ссылок они подсвечиваются красным (редактор не находит источник).

При наведении на ссылку partials с зажатым **Ctrl** она будет подсвеченa, и по нажатию на нее будет открыт файл-источник.

Рисунок 37. Ссылка для перехода



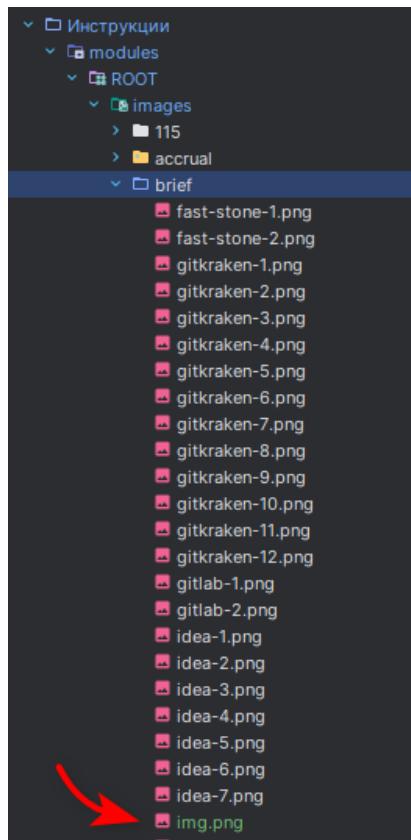
При вставке нового partials в **general** необходимо дополнительно делать коммит и Push в проекте **general** или **admin**.

Включение изображений

Для добавления изображения в документ нужно сделать следующее:

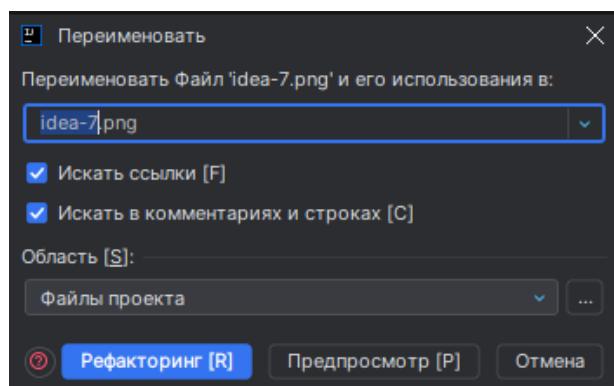
Скопировать изображение из FastStone Capture в буфер обмена **Ctrl** + **C** и вставить его в каталог **images** или вложенный в него тематический каталог **Ctrl** + **V**.

Рисунок 38. Новый файл-изображение в каталоге



Далее нужно дать файлу наименование (иначе его потом не найти). Для этого выделите в каталоге файл, который требуется переименовать, и нажмите **Shift** + **F6** или выберите в контекстном меню **Переименовать**.

Рисунок 39. Параметры переименования



Впишите необходимое имя файла и нажмите **Рефакторинг**.

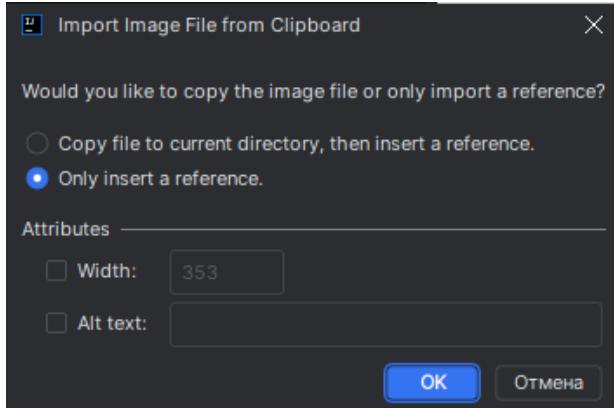
В результате имя файла будет изменено.



При изменении имени файла происходит поиск и замена ссылок на него по умолчанию в пределах проекта. Поэтому ссылки всегда будут корректны.

Для вставки в документ файл скопируйте его в каталоге **Ctrl + C** и вставьте, установив курсор в необходимое место в документе **Ctrl + V**.

Рисунок 40. Параметры вставки изображения



Не трогайте никакие параметры в диалоговом окне и просто нажмите **OK**.

В результате ссылка на файл будет помещена в указанное место и изображение будет видно в области просмотра.



Во избежание дублей, используемые во многих документах иконки и кнопки расположены в каталоге [modules/ROOT/images](#) кнопка проекта **general** и **admin** и являются своеобразной библиотекой.

Ссылка на библиотеку иконок

```
image:general::кнопка/ок.png[ ].
```

Часто используемые изображения интерфейсов находятся уровнем выше [modules/ROOT/images](#)

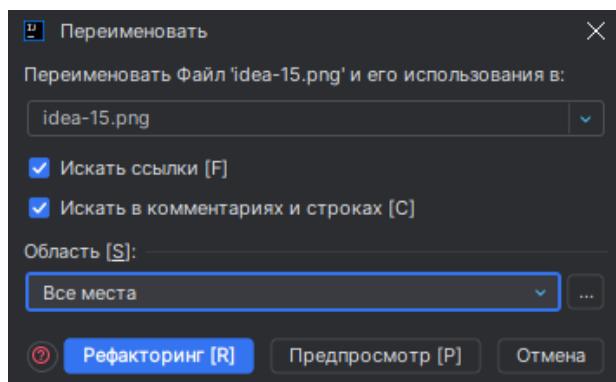


При вставке нового изображения в **general** или **admin** необходимо дополнительно делать коммит и **Push** в проекте **general**.

Рефакторинг

При переименовании источника (файла adoc или изображения) в IntelliJ IDEA происходит автоматическое обновление всех ссылок на этот файл в выбранной области.

Рисунок 41. Параметры переименования



Область выбирается из выпадающего списка соответствующего поля в модальном окне **Переименовать**:

- Все места - все файлы во всех проектах в каталоге GIT;
- Файлы проекта - все файлы текущего проекта (по расположению текущего файла).

Включение диаграмм

UML-диаграммы при необходимости вставляются в виде блоков кода (текстовым слоем на языке [PlantUML](#)).

Структура блока приведена в разделе [Синтаксис AsciiDoc. Диаграммы](#).

Завершение

По окончанию работы следует зафиксировать локальные изменения (**Commit**), а затем синхронизировать локальную версию с версией на сервере (**Push**) в GitKraken.

Затем запросить слияние (**Merge Request**) своей ветки с веткой master текущего проекта или уведомить руководителя о завершении задачи.

В результате успешного слияния и выполнения сборочной линии ваш файл появится на главной странице документации проекта и будет доступен по ссылке в форматах .html и .pdf.

Синтаксис AsciiDoc

AsciiDoc - язык разметки обычного текста, созданный как простая альтернатива XML-формату DocBook. Файлы с расширением .adoc можно создавать с помощью любого текстового редактора, а для чтения не нужны никакие другие инструменты.

Документ AsciiDoc представляет собой серию блоков, отделенных друг от друга пустыми строками.

Каждый блок имеет контекст и подразумевается в синтаксисе, но в некоторых случаях может быть объявлен явно.

Атрибуты документа

Атрибуты выставляются под заголовком документа и применяются к нему.



Атрибуты повторяются во всех документах. Не стесняйтесь разумно копировать.

Атрибуты под заголовком документа

```
:author_1: Конфигурация «ВСЕ»
:author_2: ОИ.ВСЕ.00.00.000
:author_3: v2025.11.1
:sectnums:
:sectnumlevels: 1
:sectanchors:
```

Назначение атрибутов

- **:author_1:** Конфигурация «НПО» - часть системы (отображается на титуле pdf);
- **:author_2:** ОИ.НПО.00.00.000 - шифр операции (отображается на титуле pdf);
- **:author_3:** v2025.11.1 - версия документа (отображается на титуле pdf), где:
 - 2025 - год документа,
 - 11 месяц документа,
 - версия в месяце.
- **:sectnums:** - автонумерация заголовков (разделов) (пока не используем);
- **:sectnumlevels: 1** - уровень нумерованных заголовков (разделов) (зависимость от **:sectnums:**);
- **:partnums:** - автонумерация частей (зависимость от **:sectnums:**. не используем);
- **:sectlinks:** - номера разделов - ссылки (для веба. не используем);
- **:sectanchors:** - якоря для ссылок на заголовки.



Для отмены автонумерации части заголовков нужно вставить перед заголовком - **:sectnums!:**, а для возобновления снова **:sectnums:**.

Оглавление

Для включения оглавления документа устанавливаем `toc::[]` после таблицы с историей изменений.

В результате оглавление будет собрано автоматически.

Заголовки

Уровень заголовков определяется количеством символов `=`.

Максимальное количество уровней - 6.

Уровни заголовков

```
= Заголовок документа (0 уровень)

== Заголовок раздела 1 уровень

==== Заголовок раздела 2 уровень

===== Заголовок раздела 3 уровень

===== Заголовок раздела 4 уровень

===== Заголовок раздела 5 уровень

== Заголовок раздела 1 уровень
```



При включении атрибута `:partnum:` заголовок части должен иметь уровень 0 (`=`).

Перед заголовком может быть применен атрибут `[discrete]`. «Отдельный заголовок» объявляется и оформляется аналогично названию раздела. Но в таком случае он:

- не является частью иерархии разделов;
- не может иметь дочерних разделов;
- не включается в оглавление.

Отдельный заголовок

```
[discrete]
==== Отдельный заголовок

Текст параграфа
```

Результат

```
==== Отдельный заголовок
```

Текст параграфа

Основной текст**Параграф**

Во избежание ошибок верстки, абзацы, как и все прочие элементы, должны отбиваться пустой строкой.

Параграф

Первый абзац документа.

Второй абзац документа

Варианты переноса текста в параграфах (для общего понимания)*Верстка без разделения на абзацы*

Соседние строки обычного текста при преобразовании объединяются в один абзац.

Разрывы строк, разделяющие соседние строки, не будут отображаться в выходном документе.

Вместо этого разрыв строки будет замещен пробелом.

Результат

Соседние строки обычного текста при преобразовании объединяются в один абзац.

Разрывы строк, разделяющие соседние строки, не будут отображаться в выходном документе. Вместо этого разрыв строки будет замещен пробелом.

Пример с принудительным разрывом (+) (не используем)

Соседние строки обычного текста при преобразовании объединяются в один абзац.

Разрывы строк, разделяющие соседние строки, не будут отображаться в выходном документе. +

Вместо этого разрыв строки будет замещен пробелом.

Результат

Соседние строки обычного текста при преобразовании объединяются в один абзац.

Разрывы строк, разделяющие соседние строки, не будут отображаться в выходном документе.

Вместо этого разрыв строки будет замещен пробелом.

Пример с отбивкой пустой строкой (используем)

Соседние строки обычного текста при преобразовании объединяются в один абзац. Разрывы строк, разделяющие соседние строки, не будут отображаться в выходном документе. Вместо этого разрыв строки будет замещен пробелом.

Результат

Соседние строки обычного текста при преобразовании объединяются в один абзац. Разрывы строк, разделяющие соседние строки, не будут отображаться в выходном документе.

Вместо этого разрыв строки будет замещен пробелом.



Возможно выравнивание текста в параграфах выставлением атрибутов `[.text-left]`, `[.text-center]`, `[.text-right]`. Мы это пока не используем.

Тематический разрыв

Тематический разрыв (горизонтальная линия) выставляется `---` между пустыми строками.

Установка тематического разрыва

предыдущий текст

последующий текст

Результат

предыдущий текст

последующий текст

Форматирование текста

Знак форматирования - это символ, указывающий встроенный стиль, который конвертер должен применить к тексту. Знаки форматирования идут парами.

Пара форматирования состоит из одинакового открывающего и закрывающего знака, заключающего в себя текст, который требуется стилизовать.

Мы используем пять стилевых видов:

Виды форматирования

1. Обычный
2. Курсив
3. *Полужирный*
4. *_Полужирный курсив_*
5. `Моноширинный`

Применение

1. Обычный - основной текст;
2. *Курсив* (выделение + **Ctrl** + **I**) - не важные для системы наименования (например название внешнего документа);
3. **Полужирный** (выделение + **Ctrl** + **B**) - может быть использован как «Отдельный заголовок»;
4. **Полужирный курсив** (выделение + **Ctrl** + **I** + **B**) - наименования модулей, интерфейсов, полей системы;
5. **Моноширинный** (выделение + **'**) - значения полей, алиасы объектов, текст сообщений.

Объединение в блоки

Для того чтобы параграф или другой объект оставались на нужном уровне документа используем **+** между строк объектов.

Подключение к стилю

* *_Способ выплаты_* - наименование из справочника *_Способы выплат_* (значение выбирается из выпадающего списка). По умолчанию не заполнено. Не обязательно для заполнения. Возможные значения:

** `Банк`;
** `Почта`;
** `Пусто`.
+

Если выбрано значение с наименованием `Банк`, то становится не доступным для редактирования и обнуляется значение поля *_Номер реквизита выплаты (Почта)_*.

Результат

- **Способ выплаты** - наименование из справочника **Способы выплат** (значение выбирается из выпадающего списка). По умолчанию не заполнено. Не обязательно для заполнения. Возможные значения:
 - Банк;
 - Почта;
 - Пусто.

Если выбрано значение с наименованием **Банк**, то становится не доступным для

редактирования и обнуляется значение поля **Номер реквизита выплаты (Почта)**.

Принудительный разрыв страницы

Для принудительного разрыва страницы в pdf устанавливаем <<< в нужном месте между пустыми строками.

В результате текст принудительно будет перенесен на новую страницу.

Оформление верхних и нижних индексов

Для оформления математических формул и верхних/нижних индексов используем:

Верхний индекс

$E=mc^2$

Результат

$E=mc^2$

Нижний индекс

$H_{\sim 2 \sim 0}$

Результат

H_2O

Комбинирование

$x^{2 \sim i \sim}$

Результат

x_i^2

Клавиатурные обозначения

Для отображения клавиш используется атрибут `наименование клавиши или сочетания`.

Обозначение клавиш

Глобальный поиск в проекте производится по нажатию `kbd:[Ctrl+Shift+F]`.

Результат

Глобальный поиск в проекте производится по нажатию `Ctrl + Shift + F`.

Меню (путь к операции)

В спецификациях путь к операции в системе задается приведенным ниже способом.
Возможно перейдем на этот формат в инструкциях.

Путь к операции

`menu:Модуль Договоры[Интерфейс Участник > Меню Заявления и решения ДДС > Пункт меню Добавление/редактирование заявления правопреемникам ДДС]`.

Результат

Модуль Договоры > Интерфейс Участник > Меню Заявления и решения ДДС > Пункт меню Добавление/редактирование заявления правопреемникам ДДС.

КНОПКИ И ИКОНКИ

Позднее будет внедрено размещение пиктограмм и иконок системы в документах текстовым слоем.

Вставка иконки (с использованием библиотеки иконок)

нажмите на пиктограмму `icon:fas-bank[role=icon-box]`

Результат:

Нажмите на пиктограмму .

Вставка кнопки с любым текстом

Нажмите на пиктограмму `[.btn-primary]#btn:[OK]#.`

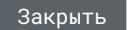
Нажмите на пиктограмму `[.btn-success]#btn:[Создать]#.`

Нажмите на пиктограмму `[.btn-secondary]#btn:[Закрыть]#.`

Результат

Нажмите на пиктограмму .

Нажмите на пиктограмму .

Нажмите на пиктограмму .

Списки

Неупорядоченный список

В начале каждого пункта неупорядоченного списка ставим * (звездочка) и пробел.

Уровень списка отбивается количеством *.

Уровни неупорядоченного списка

- * Пункт;
- * Пункт;
- ** Подпункт,
- *** Под-подпункт,
- **** Под-под-подпункт,
- ** Подпункт,
- * Пункт.

Результат

- Пункт;
- Пункт;
 - Подпункт,
 - Под-подпункт,
 - Под-под-подпункт,
 - Подпункт,
- Пункт.

Упорядоченный список

В начале каждого пункта неупорядоченного списка ставим . (точка) и пробел.

Уровень списка отбивается количеством ..

Уровни упорядоченного списка

- . Пункт;
- . Пункт;
- .. Подпункт,
- ... Под-подпункт,
- Под-под-подпункт,
- .. Подпункт,
- . Пункт.

Результат

1. Пункт;
2. Пункт;

- a. Подпункт,
 - i. Под-подпункт,
 - А. Под-под-подпункт,
 - b. Подпункт,
- 3. Пункт.

Упорядоченный контролируемый список

Порядок нумерации такого списка задается атрибутом применяемым к нему **[start=***] (цифра в атрибуте - уровень начала нумерации).

Начало списка с цифры 4

```
[start=4]
. Шаг четвертый
. Шаг пятый
. Шаг шестой
```

Результат

- 4. Шаг четвертый
- 5. Шаг пятый
- 6. Шаг шестой

Смешанный список

Смешанный список комбинируется из **.** и *****.

Смешанный список

```
. Пункт;
* Подпункт,
** Под-подпункт,
. Пункт,
.. Подпункт,
. Пункт.
```

Результат

- 1. Пункт;
 - Подпункт,
 - Под-подпункт,
- 2. Пункт,
 - а. Подпункт,
- 3. Пункт.

Разъединение списков

Для разбиения списка вставляем **//-** между пустых строк внутри списка.

Разбиение списка на две части

- . Пункт;
- . Пункт;
- . Пункт.

//-

- . Пункт;
- . Пункт;
- . Пункт.

Результат

1. Пункт;
 2. Пункт;
 3. Пункт.
-
1. Пункт;
 2. Пункт;
 3. Пункт.

Список описаний

Список описаний — это список, состоящий из одного или нескольких терминов (наборов терминов), каждый из которых имеет описание.

Используем для описания полей форм.

Разделитель пробел + **⋮⋮**.

Вложенность - 3 уровня.

Структура списка описаний

- 1 уровень :: Описание
- 2 уровень :::: Описание
- 3 уровень ::::: Описание

Результат

- 1 уровень**
- Описание

2 уровень

Описание

3 уровень

Описание

Вариант использования с «шарами» (не используем)

```
* {empty}
1 уровень :: Описание
* {empty}
2 уровень :: Описание
* {empty}
3 уровень :: Описание
```

Результат**• 1 уровень**

Описание

• 2 уровень

Описание

• 3 уровень

Описание

Вариант использования списка описаний

Следует заполнить необходимые поля формы: ::

```
* *_ИД_ФЛ_* - ИД ФЛ правопреемника ДДС;
* *_ФИО_* - Фамилия, Имя, Отчество выбранного физического лица;
* *_СНИЛС_* - номер страхового свидетельства выбранного физического лица;
* *_Дата_рождения_* - дата рождения выбранного физического лица в формате `ДД.ММ.ГГГГ`;
* *_Способ_выплат_* - способ выплаты (значение выбирается из выпадающего списка). Возможные
значения:
** Банк;
** Почта;
** Не задан.
* *_Реквизиты_выплат. Банк_* - доступно для ввода, если выбран способ выплат *_Банк_*;
* *_Реквизиты_выплат. Почта_* - доступно для ввода, если выбран способ выплат *_Почта_*;
* *_Восстановление_срока_обращения_в_судебном_порядке_* - чек-бокс.
```

Результат

Следует заполнить необходимые поля формы:

- **ИД_ФЛ** - ИД ФЛ правопреемника ДДС;
- **ФИО** - Фамилия, Имя, Отчество выбранного физического лица;

- **СНИЛС** - номер страхового свидетельства выбранного физического лица;
- **Дата рождения** - дата рождения выбранного физического лица в формате **ДД.ММ.ГГГГ**;
- **Способ выплат** - способ выплаты (значение выбирается из выпадающего списка). Возможные значения:
 - Банк;
 - Почта;
 - Не задан.
- **Реквизиты выплат. Банк** - доступно для ввода, если выбран способ выплат **Банк**;
- **Реквизиты выплат. Почта** - доступно для ввода, если выбран способ выплат **Почта**;
- **Восстановление срока обращения в судебном порядке** - чек-бокс.



Можно применить выравнивание, но мы пока не используем.

Результат с горизонтальным выравниванием

Следует заполнить необходимые поля формы:

- **ИД ФЛ** - ИД ФЛ правопреемника ДДС;
- **ФИО** - Фамилия, Имя, Отчество выбранного физического лица;
- **СНИЛС** - номер страхового свидетельства выбранного физического лица;
- **Дата рождения** - дата рождения выбранного физического лица в формате **ДД.ММ.ГГГГ**;
- **Способ выплат** - способ выплаты (значение выбирается из выпадающего списка). Возможные значения:
 - Банк;
 - Почта;
 - Не задан.
- **Реквизиты выплат. Банк** - доступно для ввода, если выбран способ выплат **Банк**;
- **Реквизиты выплат. Почта** - доступно для ввода, если выбран способ выплат **Почта**;
- **Восстановление срока обращения в судебном порядке** - чек-бокс.

Предупреждения (варнинги)

Стиль отображения предупреждения определяется назначенным типом. AsciiDoc предоставляет пять типов предупреждений, представленных следующими метками по возрастанию критичности:

- TIP: - Подсказка
- NOTE: - Замечание
- WARNING: - Внимание
- CAUTION: - Предостережение
- IMPORTANT: - Важно

Метка должна быть написана заглавными латинскими буквами и сопровождаться : (двоеточием) без пробела.

Подсказка

TIP: Редактирование также возможно из **_Карточки юридического лица_**, открываемой нажатием на гиперссылку в колонке **_Полное наименование_** в интерфейсах отображающих списки юридических лиц.

Результат



Редактирование также возможно из *Карточки юридического лица*, открываемой нажатием на гиперссылку в колонке *Полное наименование* в интерфейсах отображающих списки юридических лиц.

Такие блоки могут состоять из нескольких абзацев.

Вариант использования

[NOTE]

--

Параметр `COREWEB_DEFAULT_SMTP_HOST` - почтовый сервер для отправки сообщений (без авторизации).

Параметр `COREWEB_AUTH_SMTP_HOST` - почтовый сервер для отправки сообщений при восстановлении пароля (без авторизации).

--

Вариант использования



Параметр **COREWEB_DEFAULT_SMTP_HOST**- почтовый сервер для отправки сообщений (без авторизации).

Параметр **COREWEB_AUTH_SMTP_HOST**- почтовый сервер для отправки сообщений при восстановлении пароля (без авторизации).

Таблицы

Таблица - это блок, который может иметь дополнительные настройки, и в самом простом виде имеет только столбцы и строки.

Информация по таблицам подается для понимания их устройства и настройки.

Таблицы в редактор передаются через буфер обмена из Excel и Word. Так же существуют онлайн инструменты, например [TableConvert](#).

Строка заголовка определяется в самом начале таблицы. Колонки и строки разделяются знаком | (вертикальная черта).



Пустая ячейка задается одиночным пробелом.

Пример автоматической таблицы 3 строки 2 колонки с заголовком

```
| ===
| Колонка 1 | Колонка 2
|
| Ячейка в колонке 1, строка 1
| Ячейка в колонке 2, строка 1
|
| Ячейка в колонке 1, строка 2
| Ячейка в колонке 2, строка 2
|
| Ячейка в колонке 1, строка 3
| Ячейка в колонке 2, строка 3
| ===
```

Результат

Колонка 1	Колонка 2
Ячейка в колонке 1, строка 1	Ячейка в колонке 2, строка 1
Ячейка в колонке 1, строка 2	Ячейка в колонке 2, строка 2
Ячейка в колонке 1, строка 3	Ячейка в колонке 2, строка 3

Мы будем использовать более сложные настраиваемые таблицы.

В таблице возможны настройка колонок:

- по количеству
- по ширине;
- по выравниванию;
- по форматированию.

Автонумерация

Автоматический номер в строках таблицы задается параметром 1, где * наименование текущей таблицы.



Не пугайтесь, автонумерации используется при автоматической генерации таблиц и вручную вставлять параметр в 100500 строк не нужно.



Для подсчета количества строк в таблице используется параметр 2 в заголовке таблицы.

Таблица с автонумерацией

```
.Таблица с автонумерацией
| ===
| № | Колонка 2 | Колонка 3
| {counter:Таблица с автонумерацией}
| Ячейка в колонке 2, строка 1
| Ячейка в колонке 3, строка 1
| {counter:Таблица с автонумерацией}
| Ячейка в колонке 2, строка 2
| Ячейка в колонке 3, строка 2
| {counter:Таблица с автонумерацией}
| Ячейка в колонке 2, строка 3
| Ячейка в колонке 3, строка 3
| ===
```

Результат

Таблица 2. Таблица с автонумерацией

№	Колонка 2	Колонка 3
1	Ячейка в колонке 2, строка 1	Ячейка в колонке 3, строка 1
2	Ячейка в колонке 2, строка 2	Ячейка в колонке 3, строка 2
3	Ячейка в колонке 2, строка 3	Ячейка в колонке 3, строка 3

Выравнивание колонок

Очевидно, что для лучшего отображения, колонки таблицы должны быть выровнены.

Выравнивание задается в параметре `[cols="*"]` (для всей колонки) или в конкретной ячейке перед `|`, в том числе в заголовке.

Возможные параметры выравнивания:

- `cols="1, 6, 6"` - количество колонок (1,6,6) и их относительная ширина.
- Выравнивание по горизонтали:
 - `<` — влево (по умолчанию),
 - `>` — вправо,
 - `^` — по центру.
- Выравнивание по вертикали:
 - `.<` — верх,
 - `.>` — низ,
 - `.^` — центр.

Таблица с выравниванием

```
.Таблица с выравниванием
[cols="^1,6,6"]
|===
|№ ^|Колонка 2 ^|Колонка 3

|{counter:Таблица с выравниванием}
|Ячейка в колонке 2, строка 1
|Ячейка в колонке 3, строка 1

|{counter:Таблица с выравниванием}
|Ячейка в колонке 2, строка 2
|Ячейка в колонке 3, строка 2

|{counter:Таблица с выравниванием}
|Ячейка в колонке 2, строка 3
|Ячейка в колонке 3, строка 3
|===
```

Результат

Таблица 3. Таблица с выравниванием

№	Колонка 2	Колонка 3
1	Ячейка в колонке 2, строка 1	Ячейка в колонке 3, строка 1
2	Ячейка в колонке 2, строка 2	Ячейка в колонке 3, строка 2
3	Ячейка в колонке 2, строка 3	Ячейка в колонке 3, строка 3

Объединение ячеек

Ячейки таблицы могут быть объединены по горизонтали и вертикали.

Объединение задается в конкретной ячейке перед |:

- 2+, где 2 количество объединяемых ячеек по горизонтали;
- .2+, где 2 количество объединяемых ячеек по вертикали.

Таблица с объединением ячеек

```
.Таблица с объединением ячеек
[cols="^1,6,6"]
|===
| № ^ |Колонка 2 ^1|Колонка 3

| {counter:Таблица с объединением ячеек}
2+^ |Ячейка в колонках 2 и 3, строка 1

| {counter:Таблица с объединением ячеек}
| Ячейка в колонке 2, строка 2
.2+^ .^ |Ячейка в колонке 3, строки 2 и 3

| {counter:Таблица с объединением ячеек}
.1+ |Ячейка в колонке 2, строка 3
|===
|
```

Результат

Таблица 4. Таблица с объединением ячеек

№	Колонка 2	Колонка 3
1	Ячейка в колонках 2 и 3, строка 1	
2	Ячейка в колонке 2, строка 2	Ячейка в колонке 3, строки 2 и 3
3	Ячейка в колонке 2, строка 3	

Форматирование в ячейках

Форматирование ячеек задается (ко всей колонке) в параметре `[cols="*"]` или в конкретной ячейке перед `|`:

- `a` - ячейка AsciiDoc (поддерживается разметка, списки, ссылки и т.п.),
- `m` - ячейка monospace (код, алиас объекта и т.п.),
- `e` — курсив,
- `s` — полужирный.

Таблица с форматированием

```
.Таблица с форматированием
[cols="^1, ^6m, 6"]
| ===
| № ^ |Колонка 2 |Колонка 3

| {counter:Таблица с форматированием}
| Ячейка в колонке 2, строка 1
e| Ячейка в колонке 3, строка 1

| {counter:Таблица с форматированием}
| Ячейка в колонке 2, строка 2
s| Ячейка в колонке 3, строка 2

| {counter:Таблица с форматированием}
| Ячейка в колонке 2, строка 3
a| Ячейка в колонке 3, строка 3:

* пункт 1,
* пункт 2,
* пункт 3.
| ===
```

Результат

Таблица 5. Таблица с форматированием

№	Колонка 2	Колонка 3
1	Ячейка в колонке 2, строка 1	Ячейка в колонке 3, строка 1
2	Ячейка в колонке 2, строка 2	Ячейка в колонке 3, строка 2
3	Ячейка в колонке 2, строка 3	Ячейка в колонке 3, строка 3: <ul style="list-style-type: none"> • пункт 1, • пункт 2, • пункт 3.

Подписи к объектам

Подпись вставляется над блоком объекта в формате **.Подпись.**

Подписи применяются к следующим типам объектов:

- Изображение;
- Таблица;
- Кодовый блок;
- Пример.

В блоке цитат и универсальном блоке подпись вставляется, но обрабатывается по своему.

Для подписей возможна автонумерация счетчиком по типам объектов. Пока от нас не требуют, не включаем.

БЛОКИ

Кодовый блок

Блок для отображения исходного кода с моношириным шрифтом и подсветкой синтаксиса.

Задается символами `----` в строке в начале и конце блока, а также атрибутом `[source,]` в начале блока с добавлением наименования языка программирования или разметки после запятой без пробела.

Верстка и отображение в документе аналогичны

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

Блок-цитата

Блок для отображения цитат. В нашем случае возможно применение для указания нормативных актов и подобного.

Задается символами `(__)` в строке в начале и конце блока, а также атрибутом `[source,]` в начале блока с добавлением автора цитаты (документа) после запятой без пробела.

Подпись для блока не предусмотрена.

Цитата

`[source, Остап Бендер]`

`----`

Клиента надо приучить к мысли, что ему придётся отдать деньги. Его надо морально разоружить, подавить в нем реакционные собственнические инстинкты.

`----`

Результат

Клиента надо приучить к мысли, что ему придётся отдать деньги. Его надо морально разоружить, подавить в нем реакционные собственнические инстинкты.

— Остап Бендер

Блок-пример

В таких блоках, в данном случае, демонстрируется результат обработки синтаксиса AsciiDoc как пример формата.

Задается символами `====` в строке в начале и конце блока или атрибутом `[example]` в начале

блока.

Блок с примером

Все примеры здесь сделаны таким образом.

Результат

Все примеры здесь сделаны таким образом.

Универсальный блок

С помощью универсального блока можно построить любые конструкции.

Задается символами ****** в строке в начале и конце блока или атрибутом **[sidebar]** в начале блока.

Универсальный блок

.Любимые книжки

[sidebar]

Есть люди, которые не умеют страдать, как-то не выходит. А если уж и страдают, то стараются проделать это как можно быстрее и незаметнее для окружающих.

Любимые книжки

Есть люди, которые не умеют страдать, как-то не выходит. А если уж и страдают, то стараются проделать это как можно быстрее и незаметнее для окружающих.

Вариант использования универсального блока

.Участник находится в списке на закрытие счета

[IMPORTANT]

--

Для участника существует активное решение о закрытии счета типа *_РФ-15_* в статусе `ON_PREPARE`.

--

[.text-center]

`*ACCOUNT_CLOSE_STORAGE_CHECK*`

Входящие данные операции: ::

* Временная таблица DATA_TABLE с параметрами:

** ID_SHARER (ID и LINK.ID) - ИД основного и связанного участника

Вызывает процедуру ACCOUNT_CLOSE_STORAGE из библиотеки CHECK_LIBRARY: ::

Детализация: :::

- * Наличие у участника связанной записи в основном договоре;
- * Отсутствие у связанного участника причины закрытия `Удален. Ошибочный`;
- * Совпадение дат регистрации основного и связанного участника.

Входящие данные: :::

- * Временная таблица DATA_TABLE с параметрами:
- ** ID_SHARER_FIELD - поле с идентификатором участника.

Результат

Участник находится в списке на закрытие счета



Для участника существует активное решение о закрытии счета типа **РФ-15** в статусе **ON_PREPARE**.

ACCOUNT_CLOSE_STORAGE_CHECK

Входящие данные операции:

- Временная таблица DATA_TABLE с параметрами:
 - ID_SHARER (ID и LINK.ID) - ИД основного и связанного участника

Вызывает процедуру ACCOUNT_CLOSE_STORAGE из библиотеки CHECK_LIBRARY:

Детализация:

- Наличие у участника связанной записи в основном договоре;
- Отсутствие у связанного участника причины закрытия **Удален. Ошибочный**;
- Совпадение дат регистрации основного и связанного участника.

Входящие данные:

- Временная таблица DATA_TABLE с параметрами:
 - ID_SHARER_FIELD - поле с идентификатором участника.

Ссылки

Изображения

Изображения вставляются средствами редактора без заполнения атрибутов.

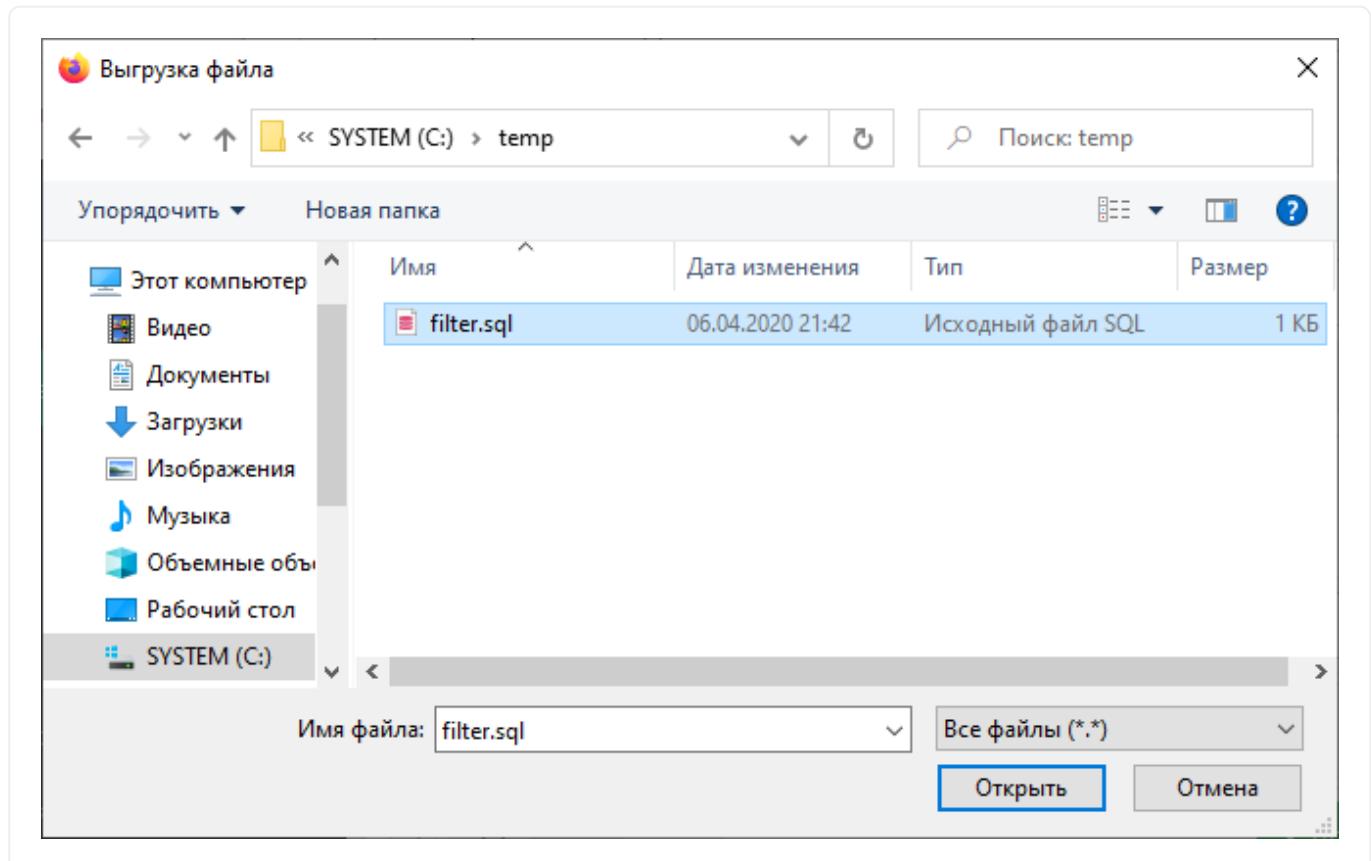


Обратите внимание, заголовки ссылок на отдельно расположенные изображения (интерфейсы, формы и подобное) (`image::`) отличаются от заголовка ссылок на изображения внутри текста (`image:`). При вставке изображений редактор автоматически распознает и вставляет корректные ссылки.

Отдельно расположенное изображение

`image::general::выгрузка-sql-2.png[]`

Результат



Изображение в тексте

Для регистрации изменений следует нажать на пиктограмму `image::кнопка/ок.png[]`.

Результат

Для регистрации изменений следует нажать на пиктограмму **Ok**.



Правила организации путей к каталогам с изображениями и размещения их в документе приведены в разделе **Порядок работы. Вставка изображений.**

Вложения (partials)

Текущий документ может включать в себя блоки (partials) с повторяющимся во многих документах текстом.

Такие блоки представляют собой автономные файлы AsciiDoc с расширением .adoc и должны быть расположены в каталоге `partials` текущего проекта.



Апострофы " ` ") в ссылках ставить не нужно. Здесь они даны для запрета рендера и отображения в финальном документе структуры ссылки.

Вложение из каталога проекта

```
`include::partial$Введение.adoc[ ]`
```

Результат

Во многих случаях partials могут быть использованы в разных проектах. Поэтому они расположены в проектах `admin` и `general`. В таком случае ссылки должны учитывать проект источника.

Вложение из каталога general

```
`include::general::partial$регистрация задания.adoc[ ]`
```

Результат

После задания параметров следует запустить задачу нажатием на пиктограмму .

В результате в очереди планировщика появится новая задача для выполнения.

При включении в документ корректной ссылки, содержимое файла источника будет отображено в правом окне редактора.



Во многих документах такой подход реализован. Смотрите, копируйте, повторяйте.

Внутренние ссылки

Перекрестные ссылки внутри документа не работают в pdf. Поэтому не используем.

Внешние ссылки

Внешние ссылки вставляются через буфер обмена с заполнением атрибутов при необходимости.

Внешние ссылки

```
https://docs.asciidoctor.org[]
https://docs.asciidoctor.org[Asciidoctor Docs]
```

Результат

```
https://docs.asciidoctor.org
Asciidoctor Docs
```

Диаграммы

В редакторе и средстве сборке документации поддерживается отображение диаграмм в текстовом слое (plantuml).

Включение диаграммы в документ производится подобно блоку кода.

Задается символами **----** в строке в начале и конце блока, а также атрибутом **[plantuml, scale=80]** в начале блока.

В коде диаграммы следует указать параметр **skinparam defaultFontName "Tahoma"**.

Атрибут

```
[plantuml, scale=80]
```

Код plantuml

```
@startuml
skinparam defaultFontName "Tahoma"
skinparam title {
FontSize 14
FontColor navy
}
title
Схема операции
"Добавление/редактирование параметров пенсии"
end title
skinparam activity {
```

```

StartColor green
EndColor tomato
BackgroundColor aliceblue
BorderColor steelblue
BorderThickness 1
}
skinparam Note {
BackgroundColor business
FontSize 9
FontColor black
}
start
:Задание параметров операции;
:Выбор участника договора;
note right
Выбор тип вкладчика договора:
- НПО ФЛ
- НПО ЮЛ
- НПО сотрудников
end note
:Выбор действия и типа решения;
note right
Выбор действия:
- добавление нового решения
- редактирование выбранного решения
- отмена выбранного решения
end note
if (Новое решение) then (да)
:Добавить в историю;
else (нет)
:Выбор назначений;
note right
Выбор зарегистрированного
решения о назначении пенсии
end note
endif
if (Добавить/редактировать) then (да)
:Ввод параметров решения;
if (НПО ЮЛ) then (да)
if (Есть вышедшие участники) then (да)
:Ввод параметров вышедших участников;
else (нет)
endif
else (нет)
endif
else (нет)
:Отмена решения;
endif
:Аудит;
:Выбор документа основания;
:Регистрация изменений;
stop

```

@enduml

Результат

**Схема операции
"Добавление/редактирование параметров пенсии"**

