

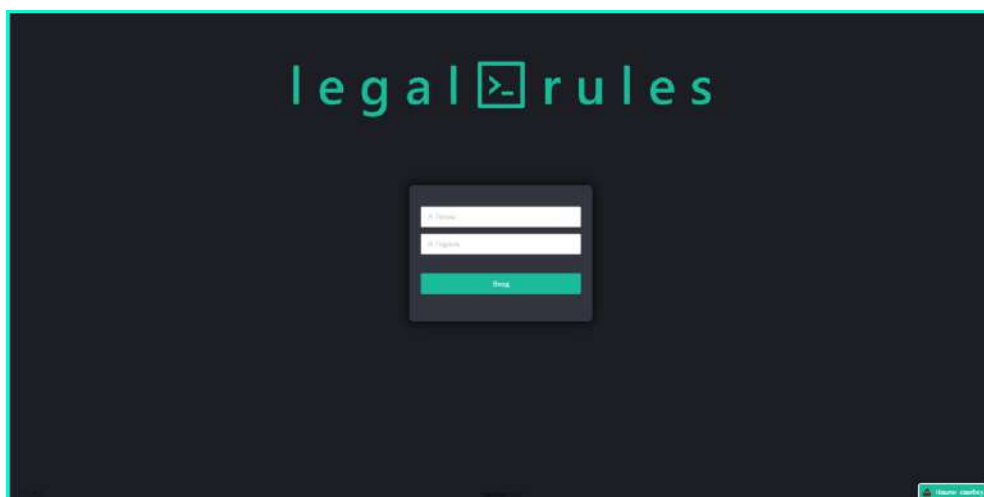
## Оглавление

|  |    |
|--|----|
| Вход в систему .....                                     | 3  |
| Главная страница .....                                   | 3  |
| Структура страницы .....                                 | 3  |
| Работа с вкладкой «Администрирование» .....              | 5  |
| Описание.....  | 5  |
| Панель «Пользователи» .....                              | 5  |
| Профиль пользователя .....                               | 7  |
| Структура профиля.....                                   | 7  |
| Пользовательские настройки.....                          | 8  |
| Добавление нового проекта .....                          | 9  |
| Работа с текущими проектами.....                         | 9  |
| Инструменты .....  | 9  |
| Управление настройками .....                             | 10 |
| Клонирование проекта .....                               | 10 |
| Работа с вкладкой «Редактирование» .....                 | 12 |
| Структура страницы .....                                 | 12 |
| Версия проекта.....                                      | 12 |
| Работа с вкладкой «Правила».....                         | 13 |
| Структура вкладки.....                                   | 13 |
| Назначение структурных элементов.....                    | 13 |
| Панель работы с кодом DSL.....                           | 19 |
| Работа с вкладкой «Глоссарий» .....                      | 31 |
| Работа с вкладкой «Поиск».....                           | 32 |
| Создание новой версии проекта .....                      | 32 |
| Управление настройками текущей версии проекта .....      | 33 |
| Структура .....  | 33 |
| Таблица правил, измененных в рамках версии проекта ..... | 33 |
| Редактирование версии проекта .....                      | 33 |
| Работа с вкладкой «Глоссарий» .....                      | 34 |
| Структура страницы .....                                 | 34 |
| Версия глоссария .....                                   | 35 |
| Создание новой версии глоссария .....                    | 35 |
| Управление настройками текущей версии глоссария.....     | 36 |
| Структура .....  | 36 |
| История изменений глоссария.....                         | 36 |

|  |    |
|--|----|
| Экспорт версии глоссария .....                               | 37 |
| Работа со структурой глоссария и панелью его элементов ..... | 37 |
| Работа с вкладкой «Плеер» .....                              | 41 |
| Описание.....  | 41 |
| Панель работы с Плеером .....                                | 42 |
| Работа с запросом (выбранным файлом).....                    | 42 |
| Способы редактирования данных. ....                          | 44 |
| Изменение запроса.....                                       | 44 |
| Работа с ответом .....                                       | 45 |
| Анализ ответа.....   | 46 |
| Сборка поставки РЮ .....                                     | 46 |
| Работа с DSL .....   | 48 |
| Базовый синтаксис.....                                       | 48 |
| Типы данных .....  | 48 |
| Хранение данных .....  | 50 |
| Ветвление .....  | 50 |
| Комментарии.....   | 51 |
| Блоки кода .....   | 51 |
| Функциональные блоки .....                                   | 51 |
| Синтаксис .....  | 52 |
| Основные (statements) .....                                  | 52 |
| Контекстные (expressions).....                               | 58 |
| Создание работающей программы на примере .....               | 68 |

## Вход в систему

Для входа необходимо ввести свой логин и пароль. Предоставление доступа и учетных данных осуществляется администратором.

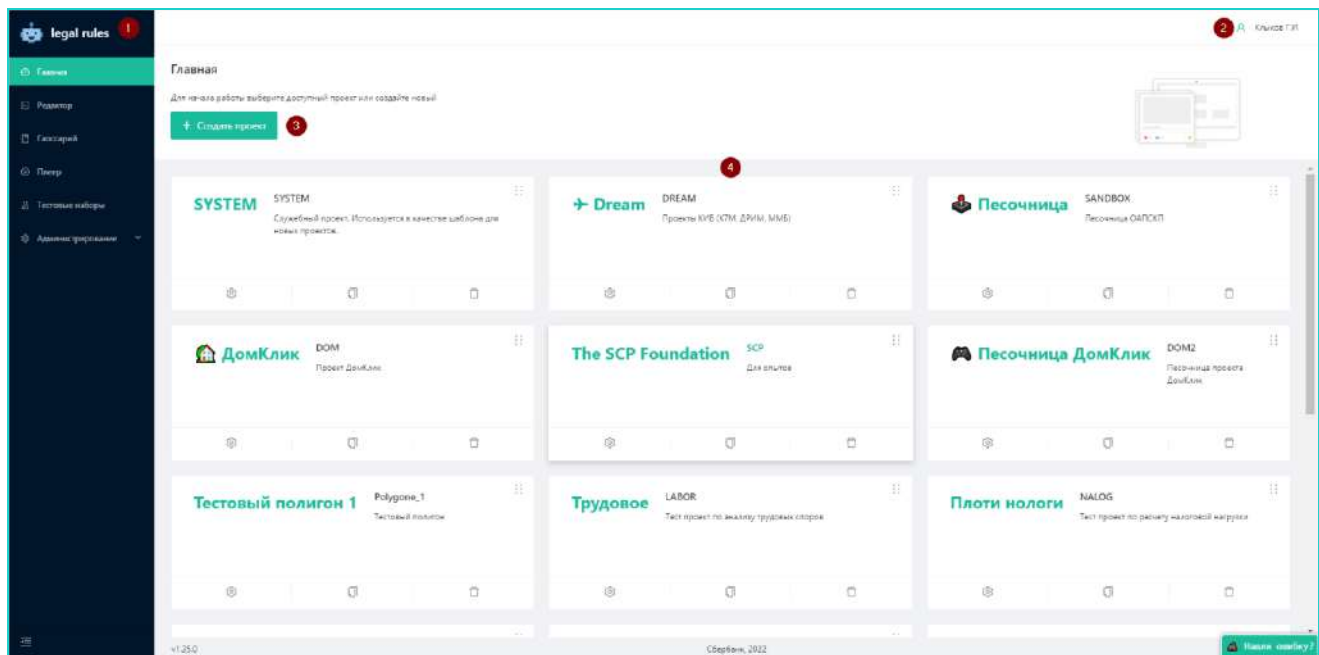


## Главная страница

### Структура страницы

После осуществления входа пользователь попадает на главную страницу ресурса, содержащую следующие элементы:

1. Панель вкладок
2. Профиль пользователя
3. Кнопка для создания проекта (доступно только для администратора)
4. Текущие проекты, доступные для пользователя



# Работа с вкладкой «Администрирование»

## Описание

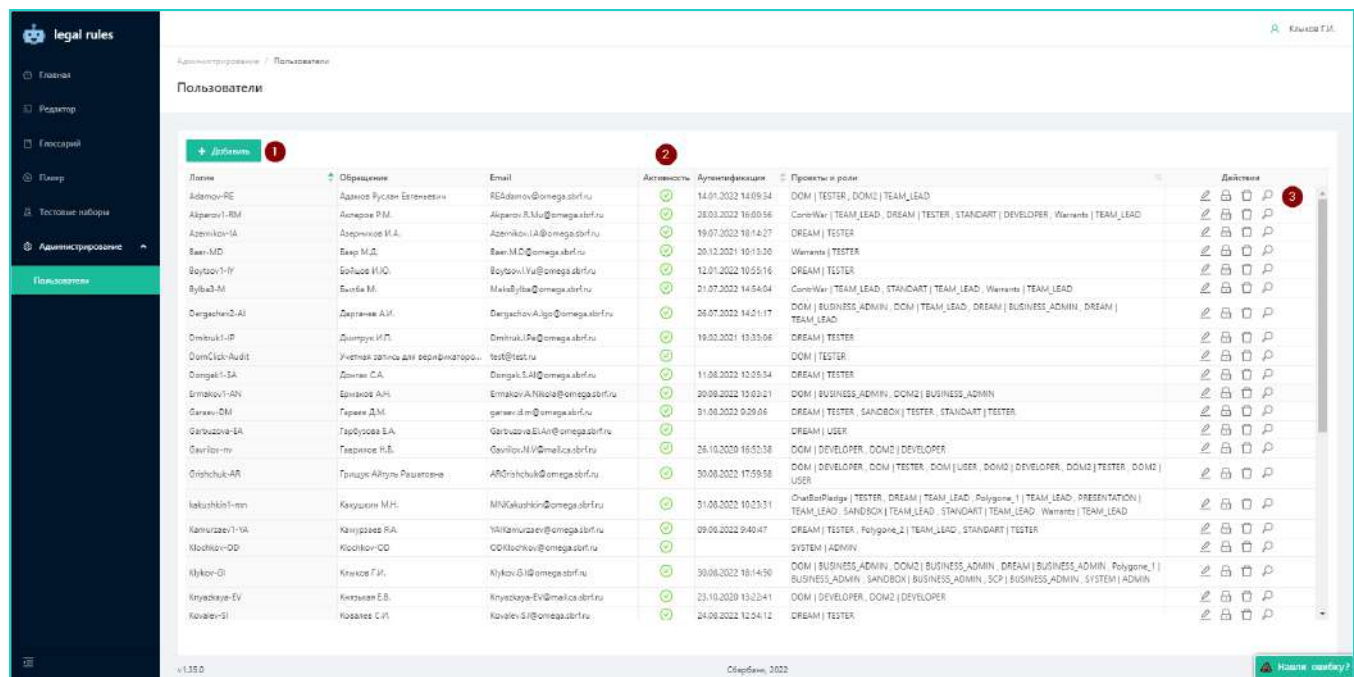
Переход на вкладку «Администрирование» осуществляется с главной страницы. Данная вкладка представляет собой инструмент управления правами и доступами пользователя. Вкладка доступна только для администратора.

## Панель «Пользователи»

## Структура

Панель «Пользователи» содержит следующие элементы:

- 1. Кнопка для добавления нового пользователя
- 2. Перечень текущих пользователей с описанием их данных и прав доступа
- 3. Инструменты редактирования списка текущих пользователей:
  - Редактирование
  - Блокировка
  - Удаление
  - Сброс пароля



## Добавление нового пользователя

Добавление пользователя осуществляется с помощью нажатия соответствующей кнопки на панели «Пользователи».

Во всплывающей панели требуется указать:

- Логин пользователя (выбирается вручную, рекомендуется формат «Фамилия-ИО»)
- Обращение для отображения в профиле пользователя
- Email пользователя
- Транспортный пароль для первичного входа (указывается вручную)

А также указать список проектов, к которым необходимо предоставить доступ и роли пользователя в них.

## Список доступных ролей

| Роль           | Доступный функционал  |
|----------------|---|
| USER           | Просмотр версий проектов, правил, глоссария.  |
| TESTER         | Все права USER<br>+<br>Доступ к функционалу тестирования правил.  |
| DEVELOPER      | Все права TESTER<br>+<br>Доступ к функционалу редактирования правил, возможность создавать версии правил, глоссария, подгружать документацию. |
| TEAM_LEAD      | Все права DEVELOPER<br>+<br>Возможность создания версий проектов, удаления версий правил в рамках версии проекта.                             |
| BUSINESS_ADMIN | Все права TEAM_LEAD<br>+<br>Возможность удаления и блокировки версий проекта, распределения прав доступа, направления сборки в ПО.            |

# Профиль пользователя

## Структура профиля

Переход в профиль пользователя осуществляется с главной страницы.

Профиль содержит следующие элементы:

1. Кнопка для изменения пароля
2. Кнопка для доступа к пользовательским настройкам
3. Информация об аккаунте
4. Перечень доступных проектов и ролей пользователя в них
5. Список заблокированных версий правил (находящихся в редактировании у пользователя)

legal rules

Профиль

1 2

Изменить пароль Пользовательские настройки

Учетная запись

Клыков Г.И.

Информация об аккаунте 3

Логин Klykov-GI

Email Klykov.GI@omega-sbfr.ru

Аутентификация 30.08.2022 16:14:59

Проекты и роли 4

DOM | BUSINESS\_ADMIN, DOM2 | BUSINESS\_ADMIN, ORBAM | BUSINESS\_ADMIN, Polygon\_1 | BUSINESS\_ADMIN, SANDBOX | BUSINESS\_ADMIN, SCP | BUSINESS\_ADMIN, SYSTEM | ADMIN

Заблокированные версии правил 5

| Название правила | Версия   | Заблокировано пользователем | Дата/время блокировки |
|------------------|----------|-----------------------------|-----------------------|
| + Тест кейс 3    | 02.10.01 | Клыков-ГИ                   | 08.04.2022 9:35:56    |
| + Тест кейс 2    | 02.10.01 | Клыков-ГИ                   | 08.04.2022 9:34:50    |
| + Тест кейс 1    | 02.10.01 | Клыков-ГИ                   | 08.04.2022 9:33:34    |
| + На разбор      | 02.01.02 | Клыков-ГИ                   | 14.09.2020 18:22:04   |

v1.25.0 Сбербанк, 2022 Написать отзыв?

## Пользовательские настройки

Пользовательские настройки содержат следующие вкладки:

- Плеер: настройка вида и формата запроса и ответа
- Редактор
- Оформление

Пользовательские настройки

Плеер Редактор Оформление

Запрос

Вид по умолчанию

Дерево

☐ Трансформировать запрос при открытии

Ответ

Вид по умолчанию

Дерево

Отмена Сохранить

Пользовательские настройки

Плеер Редактор Оформление

Редактор

☒ Автосохранение каждые

5

мин.

Размер шрифта

12

px.

Отмена Сохранить

Пользовательские настройки

Плеер Редактор Оформление

Тема

Цветовая гамма

Классическая

Меню

Расположение

Слева

Отмена Сохранить

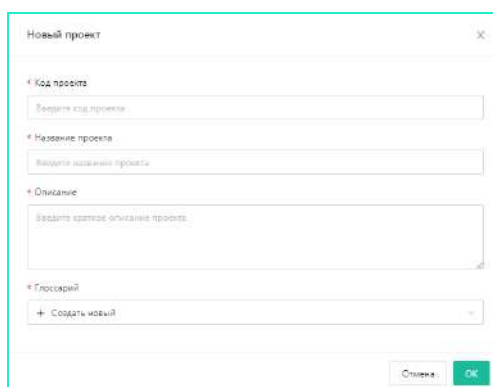


## Добавление нового проекта

Кнопка добавления проекта находится на главной странице. Кнопка доступна только для администратора.

Во всплывающем окне необходимо указать следующие параметры:

- Код проекта (только латинские буквы)
- Название проекта
- Описание проекта
- Глоссарий: есть возможность выбрать готовую структуру из списка глоссариев в существующих проектах или создать пустой глоссарий

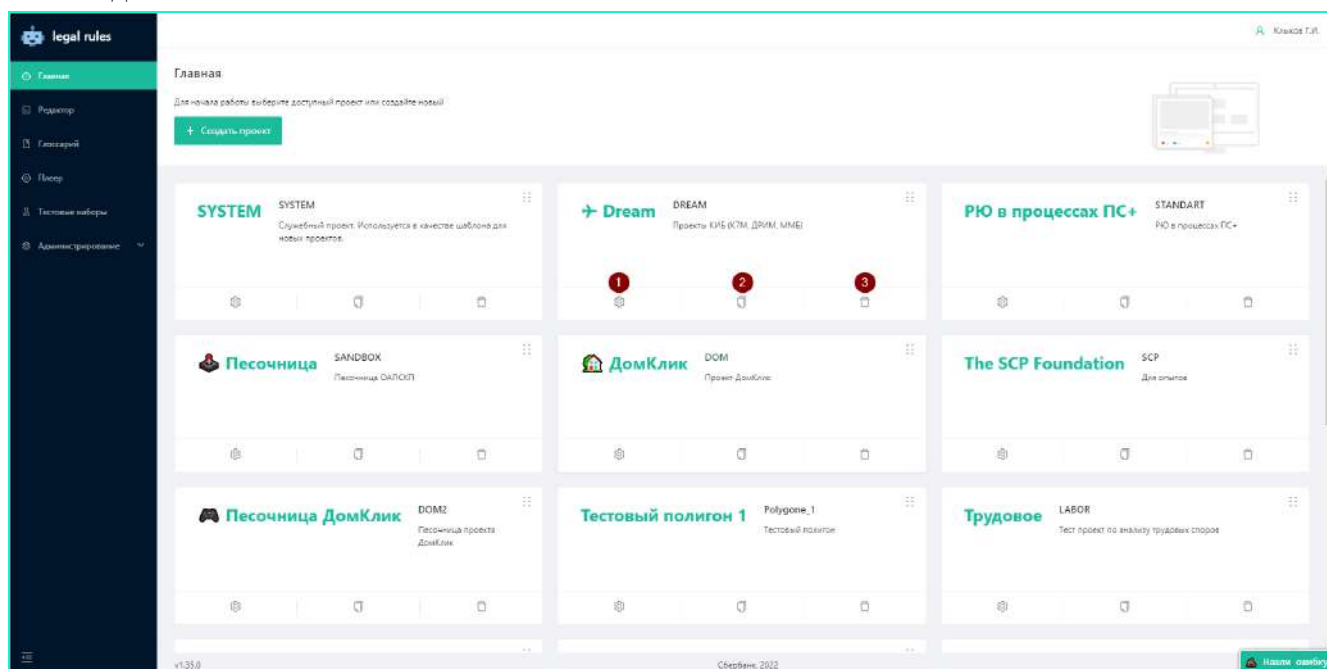


## Работа с текущими проектами

### Инструменты

С текущими проектами возможно производить следующие действия:

1. Управление настройками;
2. Клонирование;
3. Удаление.

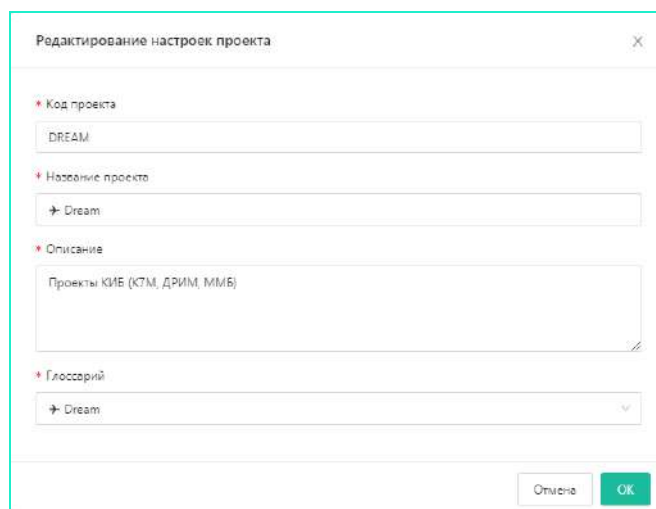


Описанные действия доступны только для администратора.

## Управление настройками

Кнопка «Настройки проекта» позволяет изменить:

- Код проекта
- Название проекта
- Описание проекта
- Глоссарий проекта



The screenshot shows a dialog box titled "Редактирование настроек проекта" (Edit Project Settings). It contains four fields, each with a red asterisk indicating it is required:

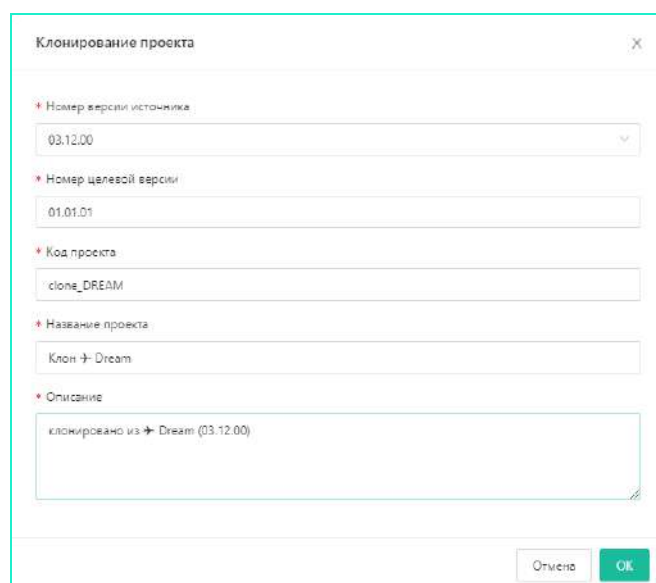
- Код проекта** (Project Code): A text input field containing "DREAM".
- Название проекта** (Project Name): A text input field containing "Dream".
- Описание** (Description): A text area containing "Проекты КИБ (К7М, ДРИМ, ММБ)".
- Глоссарий** (Glossary): A dropdown menu showing "Dream".

At the bottom right, there are two buttons: "Отмена" (Cancel) and "ОК" (OK).

## Клонирование проекта

При клонировании проекта требуется указать:

- Код проекта
- Название проекта
- Описание проекта
- Номер версии источника: какую версию проекта необходимо скопировать
- Номер целевой версии: какую версию будет иметь копируемая версия в новом проекте



The screenshot shows a dialog box titled "Клонирование проекта" (Project Cloning). It contains five fields, each with a red asterisk indicating it is required:

- Номер версии источника** (Source version number): A dropdown menu showing "03.12.00".
- Номер целевой версии** (Target version number): A text input field containing "01.01.01".
- Код проекта** (Project Code): A text input field containing "clone\_DREAM".
- Название проекта** (Project Name): A text input field containing "Клон Dream".
- Описание** (Description): A text area containing "клонировано из Dream (03.12.00)".

At the bottom right, there are two buttons: "Отмена" (Cancel) and "ОК" (OK).

При клонировании копируется версия глоссария, привязанная к указанной версии копируемого проекта.

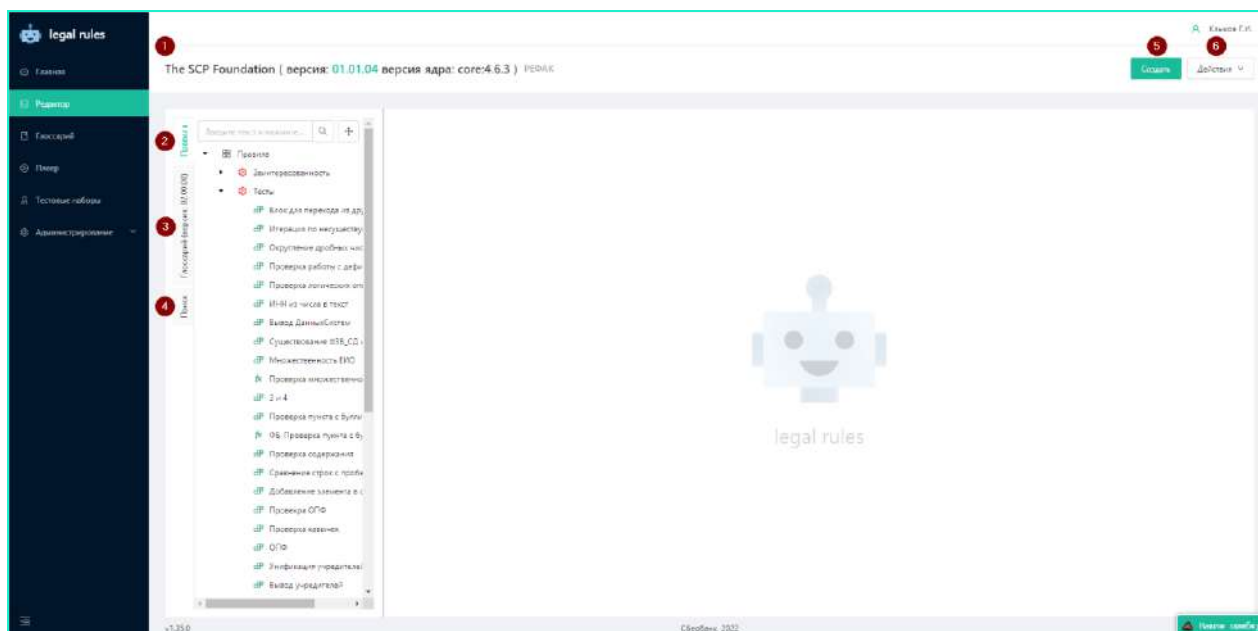


# Работа с вкладкой «Редактирование»


## Структура страницы

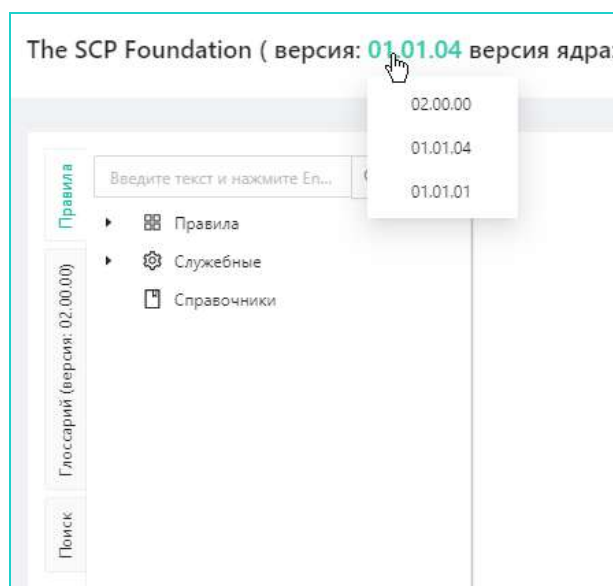
Вкладка «Редактор» содержит следующие основные элементы:

1. Описание проекта: название проекта (версия: **version** версия ядра: core) описание проекта
2. Вкладка с правилами
3. Вкладка с версией глоссария, привязанной к текущей версии проекта
4. Вкладка для поиска информации по всем правилам в текущей версии проекта
5. Кнопка для создания новой версии проекта
6. Кнопка для управления настройками текущей версии проекта



## Версия проекта

При наведении возможен выбор версии проекта. Версии со значком  являются заблокированными и не доступны для редактирования.

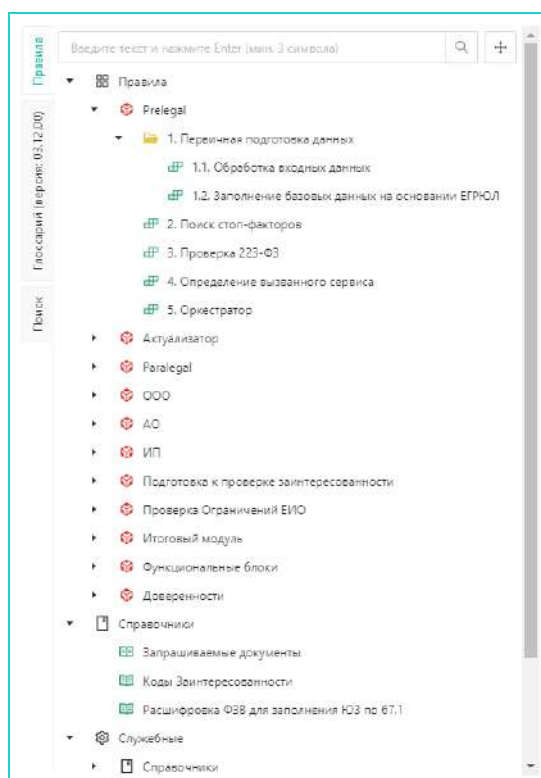


# Работа с вкладкой «Правила»<sup>1</sup>

## Структура вкладки

Вкладка представляет собой древовидную структуру, содержащую следующие структурные элементы:

- Корневой модуль:
  - Правила (📁)
  - Справочники (📖)
  - Служебные (⚙️)
- Модуль (📁)
- Каталог (📁)
- Исполняемый блок:
  - Правило (📄)
  - Функциональный блок (fx)
  - Справочник (📖)
  - Точка расширения (🔌)



## Назначение структурных элементов

### Корневой модуль «Правила»

В данном модуле размещаются основные блоки исполняемого кода на языке DSL.

Может содержать структурные элементы следующего вида:

- Модуль
- Каталог
- Исполняемый блок «Правило»

<sup>1</sup> В данном разделе взаимодействие с исполняемыми блоками описано на уровне инструмента редактирования webDSL. Взаимодействие с исполняемыми блоками на уровне кода DSL подробно описана в соответствующем разделе

- Исполняемый блок «Функциональный блок»

## Корневой модуль «Справочники»

В данном модуле размещаются пользовательские справочники.

Может содержать структурные элементы следующего вида:

- Каталог
- Исполняемый блок «Справочник»

## Корневой модуль «Служебные»

Данный модуль содержит два подмодуля:

- Справочники: технические справочники (обработка запроса/ответа, справочники для операторов DSL и т.д.)
- Точки расширения

Подмодуль «Справочники» может содержать структурные элементы следующего вида:

- Каталог
- Исполняемый блок «Справочник»

Подмодуль «Точки расширения» может содержать структурные элементы следующего вида:

- Каталог
- Исполняемый блок «Точки расширения»

## Модуль

Данный элемент используется для разделения правил на функциональные группы (разные этапы проверки, разные шаги процесса и т.д.).

Может содержать структурные элементы следующего вида:

- Каталог
- Исполняемый блок «Правило»
- Исполняемый блок «Функциональный блок»

## Каталог

Каталог используется для визуальной группировки исполняемых блоков.

Может содержать структурные элементы следующего вида:

- Каталог
- Исполняемый блок «Правило»
- Исполняемый блок «Функциональный блок»
- Исполняемый блок «Справочник»
- Исполняемый блок «Точки расширения»

## Исполняемый блок «Правила»

Основной блок, в котором осуществляется написания кода на языке DSL.

## Исполняемый блок «Функциональный блок»

Блок с кодом на языке DSL, вызов которого осуществляется в блоках правил (по аналогии с функцией в языках программирования).

## Исполняемый блок «Справочник»

Редактируемая таблица с данными для вызова в блоках правил и функциональных блоках.

## Исполняемый блок «Точка расширения»

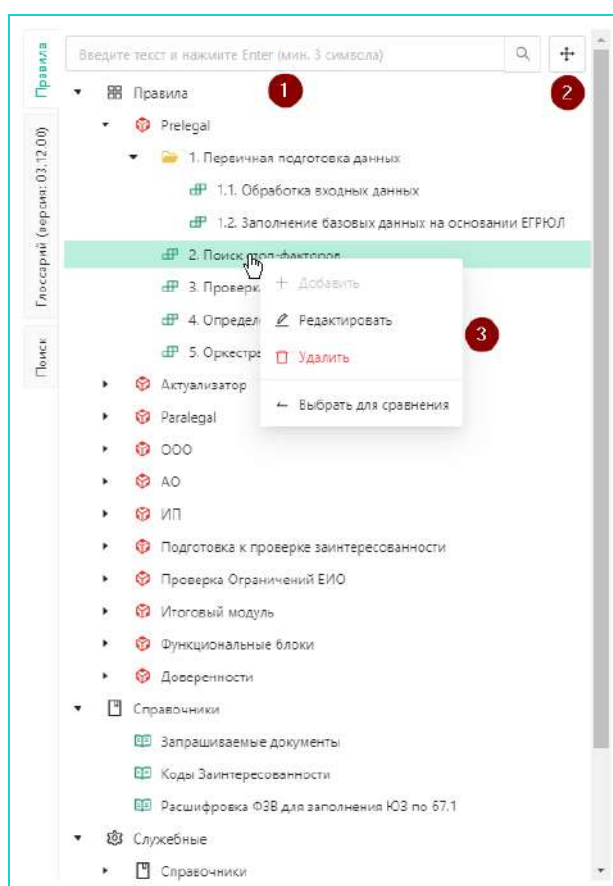
Блок с кодом на языке Python для расширения возможностей ядра и его кастомизации под нужды проекта (например, особая обработка запроса/ответа, создание/переопределение оператора и т.д.).

## Инструменты работы с деревом

### Описание

Доступны следующие инструменты работы:

1. Панель поиска
2. Кнопка подключения возможности перетаскивания элементов дерева (Drag-and-drop)
3. Всплывающая панель инструментов



## Панель инструментов

Панель инструментов открывается при нажатии правой кнопкой мыши на структурном элементе дерева и содержит следующие элементы:

- Добавить
- Редактировать
- Удалить

**! Не рекомендуется удалять версию проекта, если она используется в следующих версиях проекта**

- Выбрать для сравнения (доступно только для исполняемых блоков)

## Добавление структурных элементов

Корневые модули добавляются на уровне системы при создании проекта, на уровне пользователя возможность добавления отсутствует.

Модуль может быть создан только от корневого модуля «Правила». Для создания модуля необходимо выбрать вкладку «Каталог» и заполнить название создаваемого модуля.

Каталог может быть создан от любого структурного элемента, кроме исполняемого блока и корневых модулей «Правила» (превращается в модуль) и «Служебные» (запрещено). Для создания каталога необходимо выбрать вкладку «Каталог» и заполнить название создаваемого каталога.

Новый элемент

Тип элемента

Каталог Правило

\* Наименование

Введите наименование

Наименование файла

Введите наименование файла

\* Краткое наименование ⓘ

Введите краткое наименование

Отмена ОК

Исполняемый блок типа «Правило» может быть создан только в корневом модуле «Правила». Для создания блока необходимо в корневом модуле/модуле/каталоге, выбрать вкладку «Правило» и заполнить следующие поля:

- Наименование: наименование создаваемого правила
- Тип правила: «основной»
- Язык: «DSL»

Новый элемент

Тип элемента

Каталог Правило

\* Наименование

Введите наименование

\* Тип правила

основной

\* Язык

DSL

Наименование файла

Введите наименование файла

Отмена ОК



Исполняемый блок типа «Функциональный блок» может быть создан только в корневом модуле «Правила». Рекомендуется создавать блок в модуле «Функциональные блоки». Для создания блока необходимо в корневом модуле/модуле/каталоге, выбрать вкладку «Правило» и заполнить следующие поля:

- Наименование: наименование создаваемого функционального блока
- Тип правила: «функциональный»
- Язык: «DSL»
- Краткое наименование для вызова функционального блока

The screenshot shows a dialog box titled 'Новый элемент' (New Element) with a close button (X) in the top right corner. It contains the following fields and options:

- Тип элемента:** Two buttons, 'Каталог' (Catalog) and 'Правило' (Rule), with 'Правило' selected.
- \* Наименование:** A text input field with the placeholder 'Введите наименование' (Enter name).
- \* Тип правила:** A dropdown menu with 'функциональный' (functional) selected.
- \* Язык:** A dropdown menu with 'DSL' selected.
- Наименование файла:** A text input field with the placeholder 'Введите наименование файла' (Enter file name).
- \* Краткое наименование ⓘ:** A text input field with the placeholder 'Введите краткое наименование' (Enter short name).
- At the bottom are two buttons: 'Отмена' (Cancel) and 'ОК' (OK).

Исполняемый блок типа «Справочник» может быть создан только в корневом модуле «Справочники» или в каталоге «Справочники» в корневом модуле «Служебные». Для создания блока необходимо в корневом модуле/каталоге выбрать вкладку «Правило» и заполнить следующие поля:

- Наименование: наименование создаваемого справочника;
- Краткое наименование для вызова справочника.

The screenshot shows a dialog box titled 'Новый элемент' (New Element) with a close button (X) in the top right corner. It contains the following fields and options:

- Тип элемента:** Two buttons, 'Каталог' (Catalog) and 'Правило' (Rule), with 'Правило' selected.
- \* Наименование:** A text input field with the placeholder 'Введите наименование' (Enter name).
- Наименование файла:** A text input field with the placeholder 'Введите наименование файла' (Enter file name).
- \* Краткое наименование ⓘ:** A text input field with the placeholder 'Введите краткое наименование' (Enter short name).
- At the bottom are two buttons: 'Отмена' (Cancel) and 'ОК' (OK).

Исполняемый блок типа «Точка расширения» может быть создан только в каталоге «Точки расширения» в корневом модуле «Служебные». Для создания блока необходимо в корневом модуле/каталоге выбрать вкладку «Правило» и заполнить следующие поля:

- Наименование: наименование создаваемой точки расширения
- Тип правила: «точка расширения»
- Краткое наименование для вызова точки расширения

The screenshot shows a dialog box titled 'Новый элемент' (New Element) with a close button (X) in the top right corner. It contains the following fields and options:

- Тип элемента:** Two buttons, 'Каталог' (Catalog) and 'Правило' (Rule), with 'Правило' selected.
- \* Наименование:** A text input field with the placeholder 'Введите наименование' (Enter name).
- \* Тип правила:** A dropdown menu with 'точка расширения' (extension point) selected.
- Наименование файла:** A text input field with the placeholder 'Введите наименование файла' (Enter file name).
- \* Краткое наименование ⓘ:** A text input field with the placeholder 'Введите краткое наименование' (Enter short name).
- At the bottom are two buttons: 'Отмена' (Cancel) and 'ОК' (OK).

## Редактирование структурных элементов

При редактировании необходимо указать новое наименование структурного элемента. При редактировании наименования исполняемого блока в дереве наименование внутри блока правил не меняется.

**! При редактировании наименования модуля необходимо изменить соответствующее наименование для оператора «Перейти в модуль».**

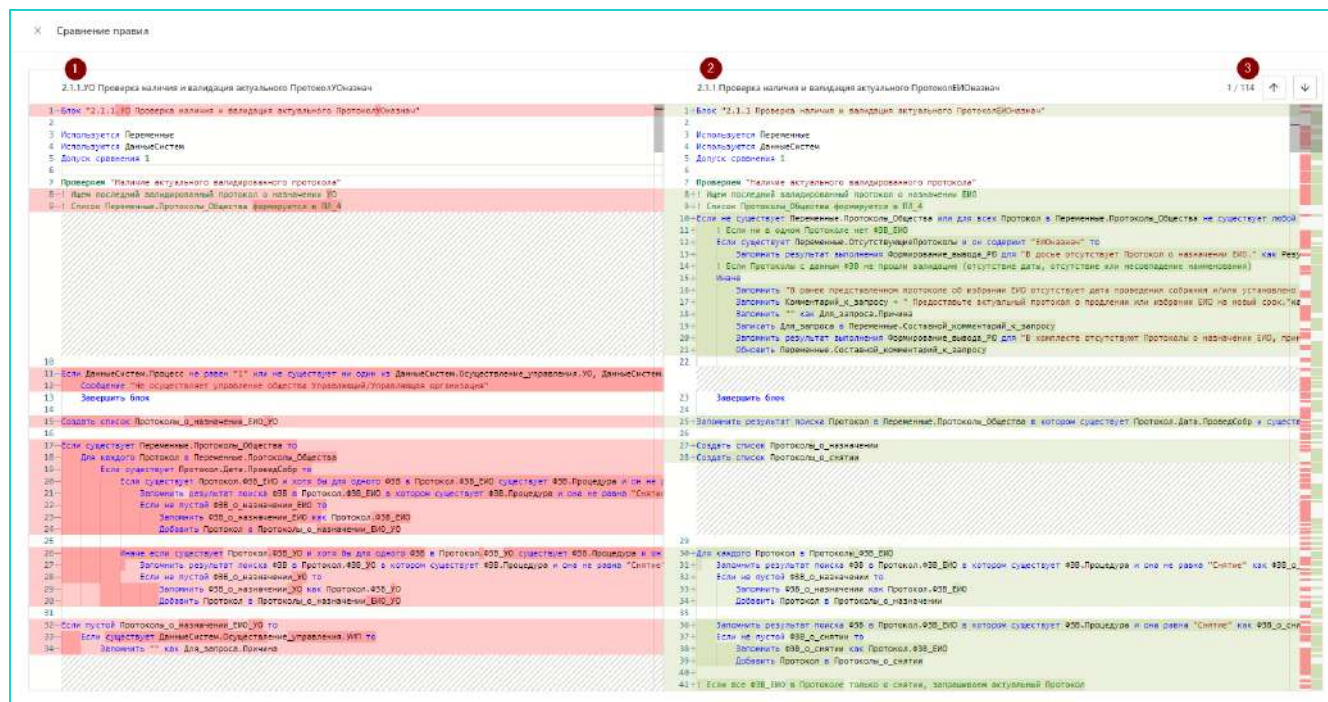
## Сравнение двух исполняемых блоков

Для сравнения двух блоков необходимо:

1. Нажать правой кнопкой мыши на блок, который будет базой сравнения
2. Выбрать опцию «Выбрать для сравнения»
3. Нажать правой кнопкой мыши на блок, с которым необходимо произвести сравнение
4. Выбрать опцию «Сравнить» (появится под опцией «Выбрать для сравнения»)

В результате откроется панель с результатом сравнения, содержащая следующие элементы:

1. Название блока – базы для сравнения с перечнем устаревших данных (красная заливка)
2. Название блока для сравнения с перечнем обновленных данных (зеленая заливка)
3. Счетчик изменений и кнопки переключения между ними

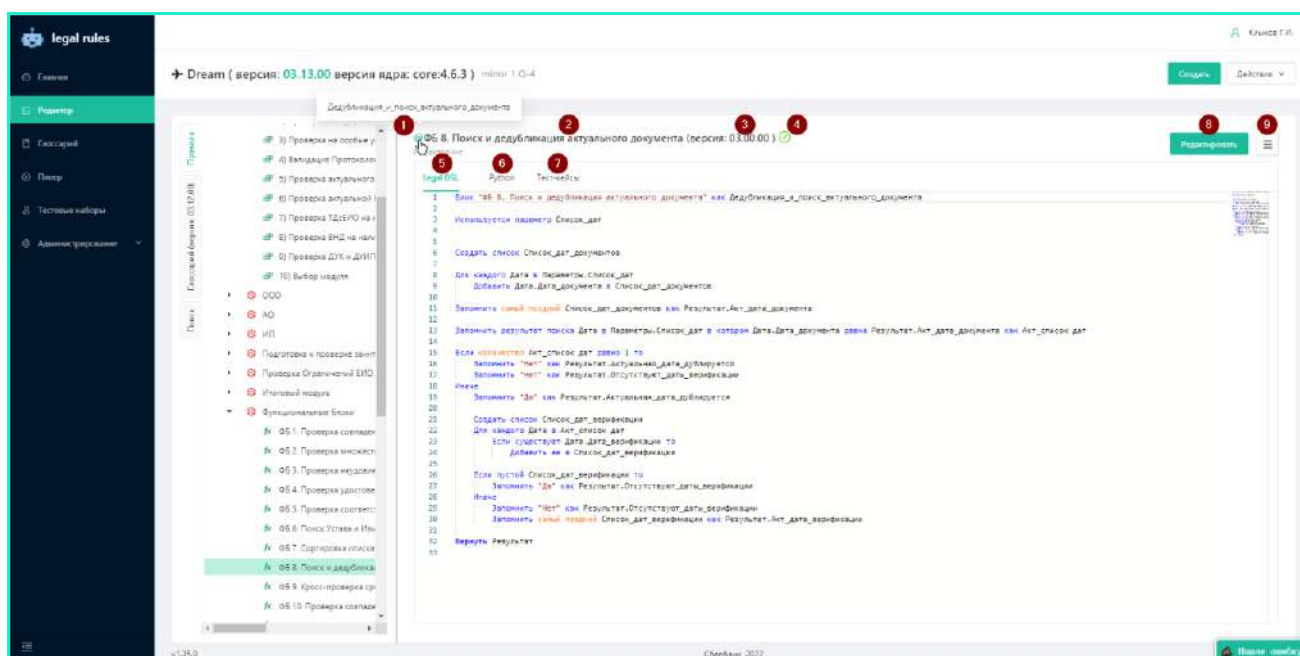


## Панель работы с кодом DSL

### Панель для исполняемых блоков типа «Правило» и «Функциональный блок» Структура

Панель открывается при выборе исполняемого блока и содержит следующие элементы:

1. Только для функционального блока: метка функционального блока, при наведении отображается краткое наименование для его вызова
2. Наименование блока: меняется в настройках блока и не связано с наименованием блока в дереве
3. Версия блока правил
4. Метка акцепта изменений в версии правил (Ревью)
5. Вкладка с кодом на языке DSL
6. Вкладка с преобразованным кодом на языке Python
7. Вкладка с тест-кейсами
8. Кнопка редактирования блока правил
9. Панель управления настройками блока правил



## Вкладка «Тест-кейсы»

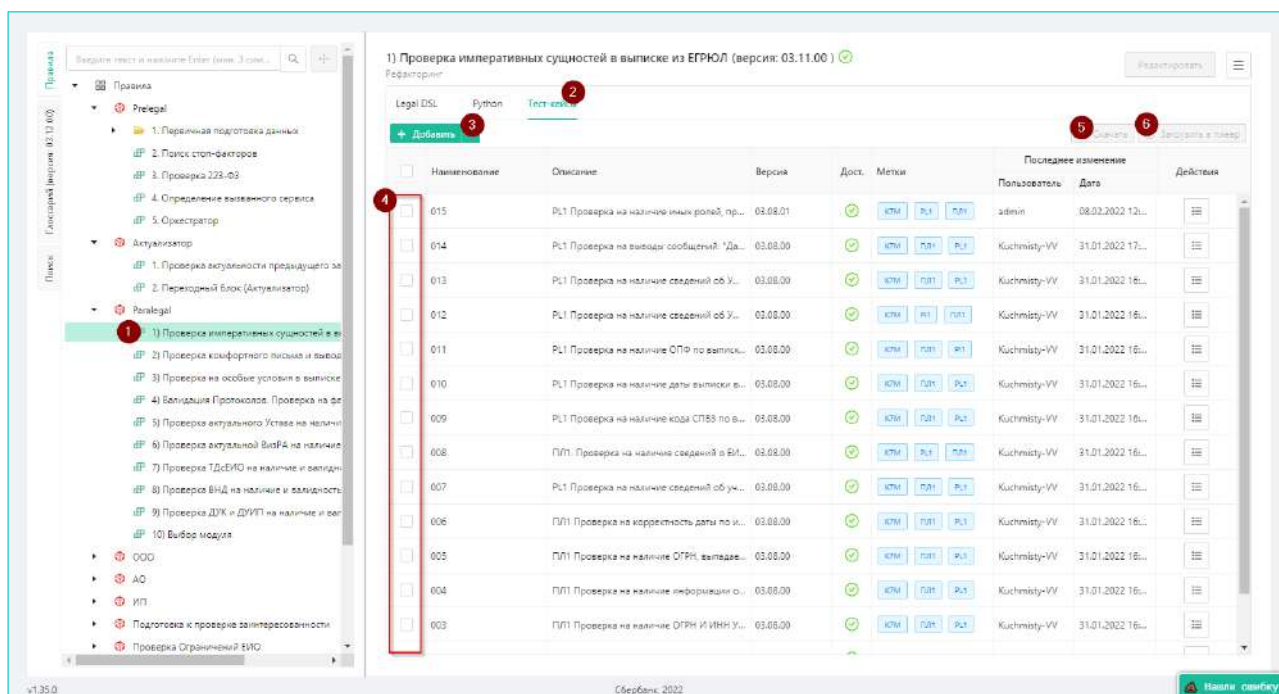
### Описание

Вкладка «Тест-кейсы» представляет собой хранилище файлов для тестирования (тест-кейсов) конкретного блока правил. Тест-кейсы хранятся в блоках, для которых были созданы.

### Структура страницы

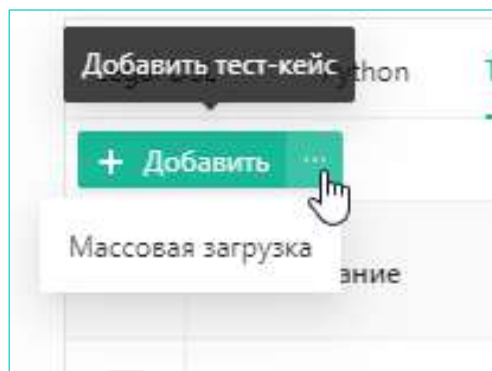
Вкладка с тест-кейсами имеет следующую структуру:

1. Выбор нужного блока правил для работы
2. Вкладка с тест-кейсами
3. Кнопка для добавления файлов
4. Выбор добавленных тест-кейсов
5. Кнопка для скачивания выбранных кейсы
6. Кнопка для загрузки выбранных кейсов в плеер для тестирования



### Добавление файлов

Кнопка «Добавить» позволяет произвести массовую загрузку тест-кейсов.



Алгоритм загрузки выглядит следующим образом:

1. Ставим необходимые метки (нужны для дальнейшей сортировки и отбора кейсов, а также упрощают их поиск)
2. Добавляем файлы в соответствующую область
3. Кликаем «ОК»

Новый тест-кейс

Префикс

Введите префикс для имени тест-кейса

Метки

+ Новый элемент

Нажмите или перенесите файлы в данную область.  
Поддерживается единичная или массовая загрузка JSON файлов

Отмена ОК

## Работа с кейсом

Таблица с тест-кейсами имеет следующую структуру:

- Наименование
- Описание
- Версия: версия правила, для которой был создан тест-кейс
- Дост. (доступность): возможность использования тест-кейса не только его создателем, но и другими пользователями
- Метки
- Последнее изменение: логин пользователя и дата
- Действия

| Legal DSL Python Тест-кейсы |                      |                           |          |       |                  |                     |                     |          |
|-----------------------------|----------------------|---------------------------|----------|-------|------------------|---------------------|---------------------|----------|
| + Добавить                  |                      | Скачать Загрузить в плеер |          |       |                  |                     |                     |          |
| <input type="checkbox"/>    | Наименование         | Описание                  | Версия   | Дост. | Метки            | Последнее изменение |                     | Действия |
|                             |                      |                           |          |       |                  | Пользователь        | Дата                |          |
| <input type="checkbox"/>    | Выбранный файл(кейс) |                           | 03.13.00 | ✓     | Тест Тест1 Блок1 | Garaev-DM           | 08.09.2022 18:21:19 |          |

Для быстрого редактирования полей «Наименование» и «Описание» необходимо:

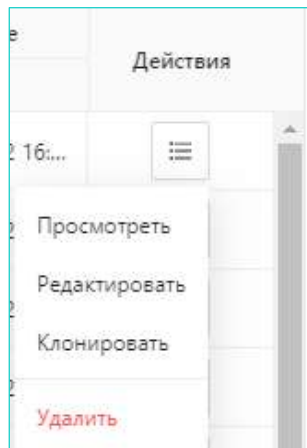
1. Дважды кликнуть левой кнопкой мыши на поле «Наименование»/ «Описание»
2. Внести необходимые исправления
3. Сохранить путем нажатия на галочку в поле «Действия»

| Legal DSL Python Тест-кейсы |                      |                           |          |       |  |                     |                  |          |
|-----------------------------|----------------------|---------------------------|----------|-------|--|---------------------|------------------|----------|
| + Добавить                  |                      | Скачать Загрузить в плеер |          |       |  |                     |                  |          |
| <input type="checkbox"/>    | Наименование         | Описание                  | Версия   | Дост. | Метки  | Последнее изменение |                  | Действия |
|                             |                      |                           |          |       |  | Пользователь        | Дата             |          |
| <input type="checkbox"/>    | Выбранный файл(кейс) |                           | 03.13.00 | ✓     | Тест X Тест1 X<br>Блок1 X<br>+ Новый элемент | Garaev-DM           | 08.09.2022 15... | ✓ X      |

## Действия с тест-кейсом

Нажатие на кнопку «Действия» открывает следующий функционал:

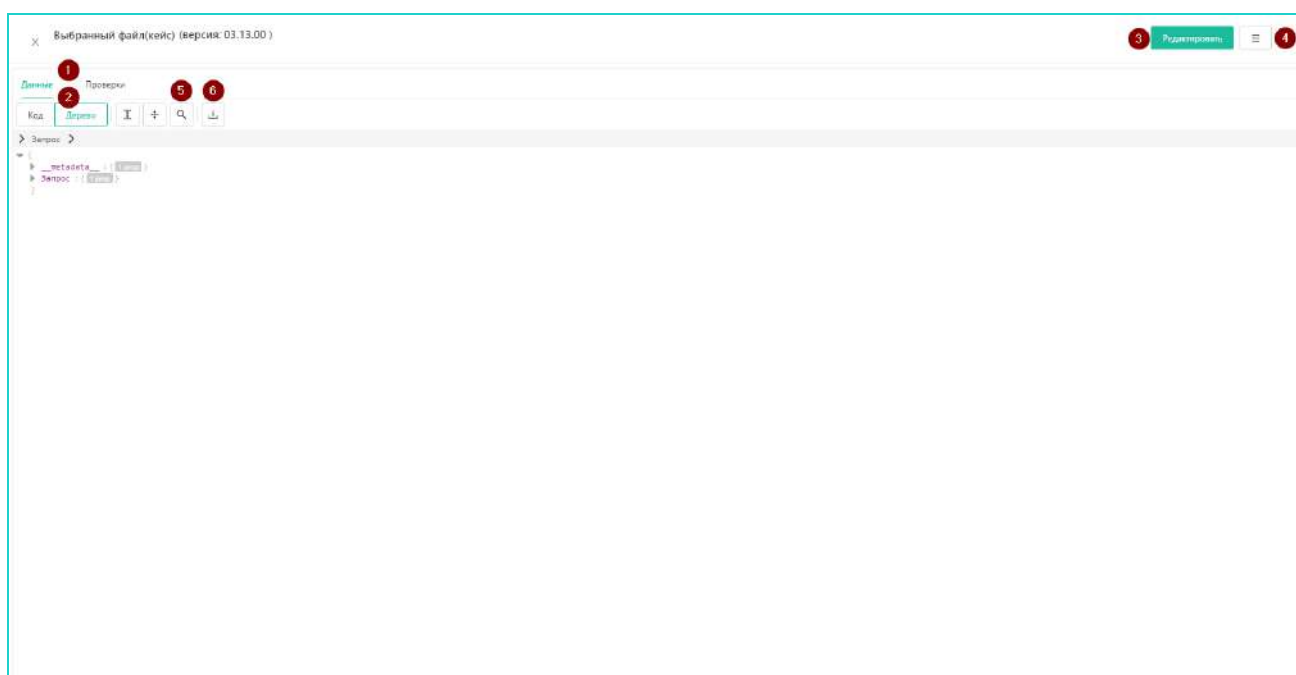
- Просмотреть: переход в структуру тест-кейса для просмотра содержимого файла
- Редактировать: внесение изменений в наименование и описание тест-кейса
- Клонировать: создание дубликата тест-кейса
- Удалить



## Действие «Просмотреть»

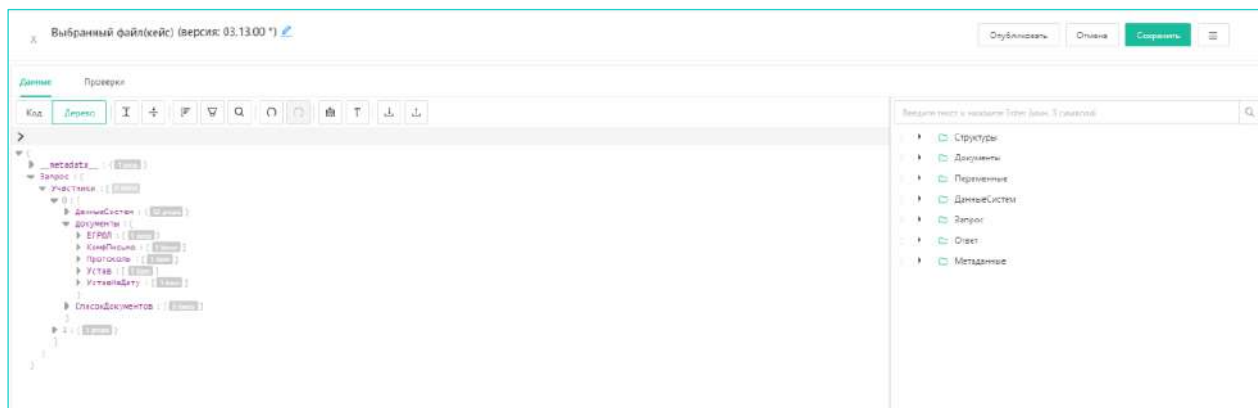
Нажатие на кнопку осуществляет переход в структуру тест-кейса для просмотра содержимого файла:

1. Вкладка с содержимым тест-кейса
2. Формат отображения тест-кейса
3. Кнопка редактирования
4. Кнопка с панелью управления, содержащей следующие элементы:
  - Свойства
  - Удаление тест-кейса
5. Поиск по тексту
6. Загрузка тест-кейса на локальный компьютер

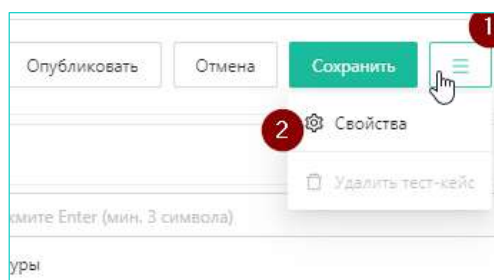


## Редактирование тест-кейса

При нажатии на кнопку «Редактировать» появляется возможность изменять, добавлять, удалять данные в запросе (функционал редактирования подробно описан в главе «Работа с вкладкой «Плеер»»).

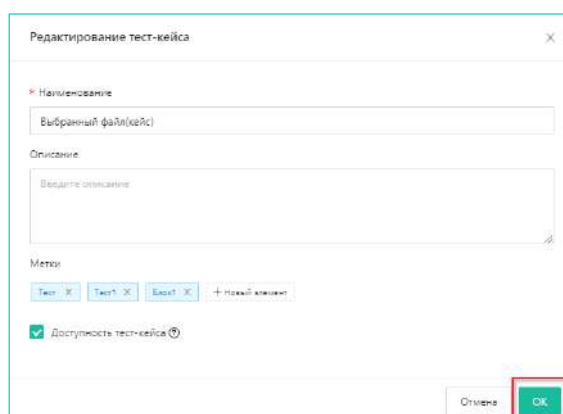


Для изменения наименования, описания, меток внутри запроса, необходимо выбрать раздел «Свойства» на панели управления тест-кейсом.



Появляется окно, позволяющее изменить следующие элементы:

- Наименование
- Описание
- Метки
- Доступность



После внесения изменений необходимо:

1. Опубликовать изменения
2. Внести комментарии изменения
3. Сохранить тест-кейс



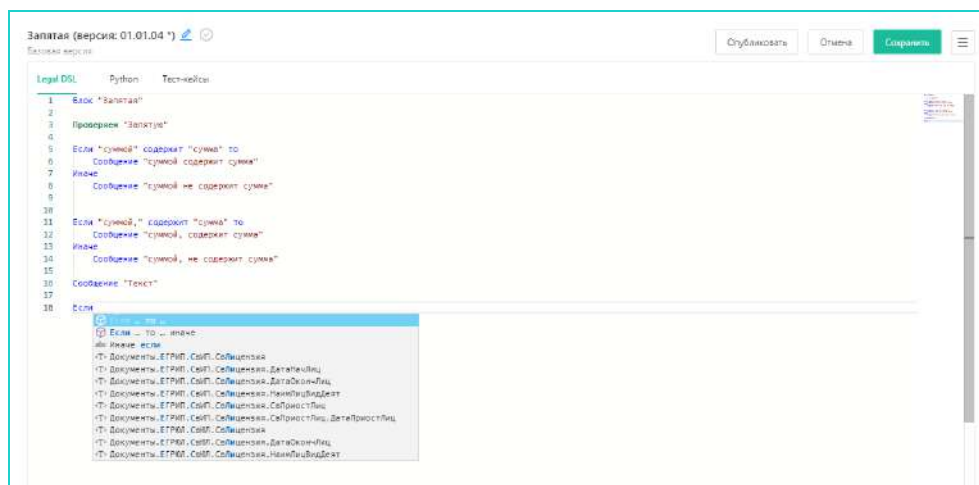
## Редактирование блока

Блок переходит в режим редактирования при нажатии кнопки «Редактировать».

При этом:

- Появляется возможность изменения кода на языке DSL
- Создается версия правила под текущую версию проекта (если не была создана ранее)
- Снимается метка акцепта
- Вместо кнопки «Редактировать» появляются следующие кнопки:
  - Опубликовать
  - Отмена
  - Сохранить

Панель редактирования снабжена подсказчиком (аналог IntelliSense), помогающим в написании синтаксиса операторов, а также путей элементов глоссария. Также возможно использование горячих клавиш: копирование – Ctrl+C, вставка – Ctrl+V, поиск по тексту и замена – Ctrl+F, поиск по номеру строки – Ctrl+N и т.д.

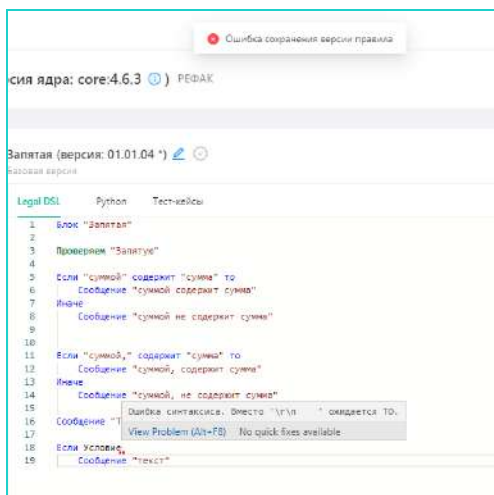


При наличии более поздней версии правила появится предупреждающий значок.



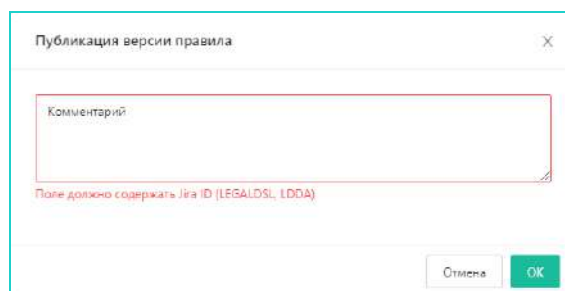
При нажатии кнопки «Отмена» код и все настройки блока правил возвращаются в состояние, в котором они были до редактирования.

При нажатии кнопки «Сохранить» происходит автоматическая конвертация кода на языке DSL в код на языке Python. При выявлении ошибок конвертации будет выведено предупреждение с указанием местоположения ошибки. В случае успешного сохранения будет выведено сообщение «Вы великолепны!».





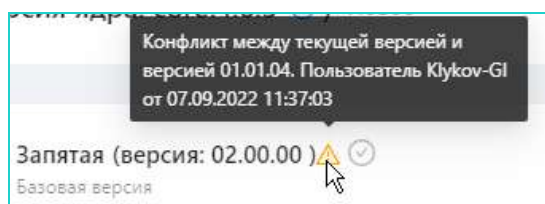
При нажатии кнопки «Опубликовать» происходит конвертация кода по аналогии с «Сохранить», а также всплывает окно, в котором необходимо указать комментарий к публикуемому изменению, а также номер задачи Jira из проектной области, указанной в настройках проекта.



Если проектная область не указана, номер задачи указывать необязательно.

Пользователь с ролью администратора может проставить метку акцепта, нажав на нее. В этом случае указывать номер задачи из Jira также не требуется.

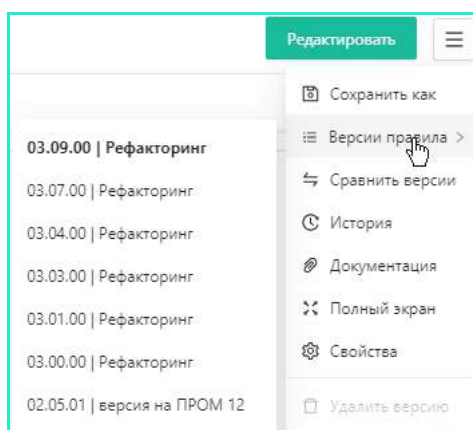
Если у публикуемой версии правила уже создана более старшая версия, то в последней происходит автоматическое снятие метки акцепта и проставление предупреждения.



## Управление настройками блока

Кнопка управления настройками содержит следующие элементы:

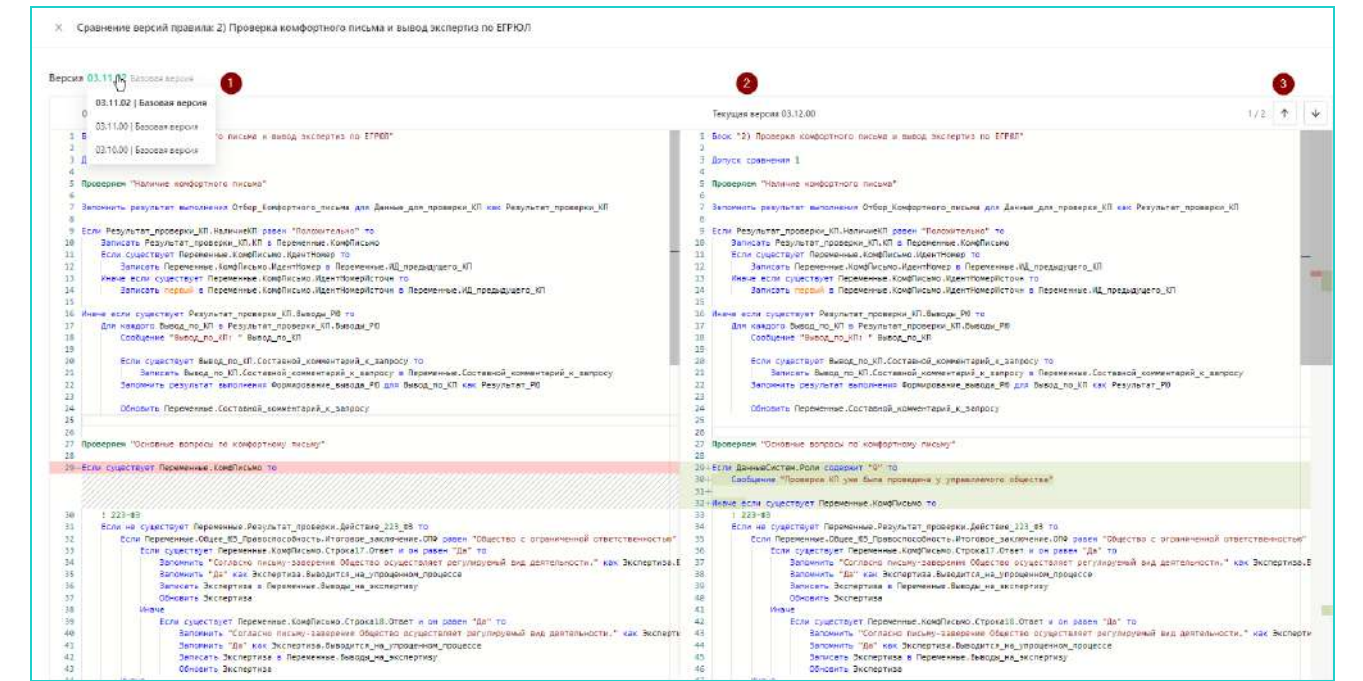
- Сохранить как
- Версии правила: просмотр версий правила в других версиях проекта
- Сравнить версии: сравнение текущей версии правила с предыдущей
- История: сравнение двух опубликованных версий правила
- Документация: вложения для версии правила
- Полный экран: раскрытие вкладки на весь экран
- Свойства: просмотр и редактирование версии правила
- Удалить версию: удаление текущей версии правила (недоступно удаление версии правила из другой версии проекта)



# Сравнение версий

Функционал сравнения версий позволяет сравнить правило в текущей и в предыдущей версии проекта и содержит следующие элементы:

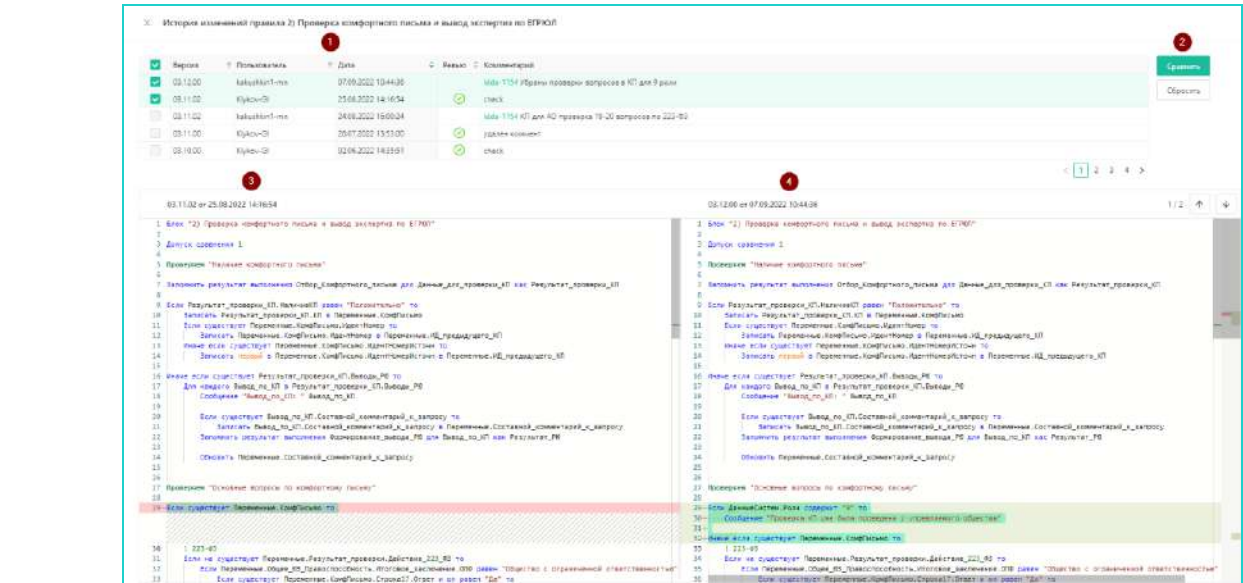
- 1. Выбранную предыдущую версию проекта с перечнем устаревших данных (красная заливка)
- 2. Текущую версию проекта с перечнем обновленных данных (зеленая заливка)
- 3. Счетчик изменений и кнопки переключения между ними



# История изменений

Функционал истории изменений позволяет сравнить две любые публикации правила, как в рамках одной версии проекта, так и в рамках разных, и содержит следующие элементы:

- 1. Панель выбора сравниваемых публикаций с возможностью фильтрации и сортировки
- 2. Кнопки для запуска сравнения и сброса выбранных публикаций
- 3. Публикацию – базу для сравнения с перечнем устаревших данных (красная заливка)
- 4. Публикацию для сравнения с перечнем обновленных данных (зеленая заливка)

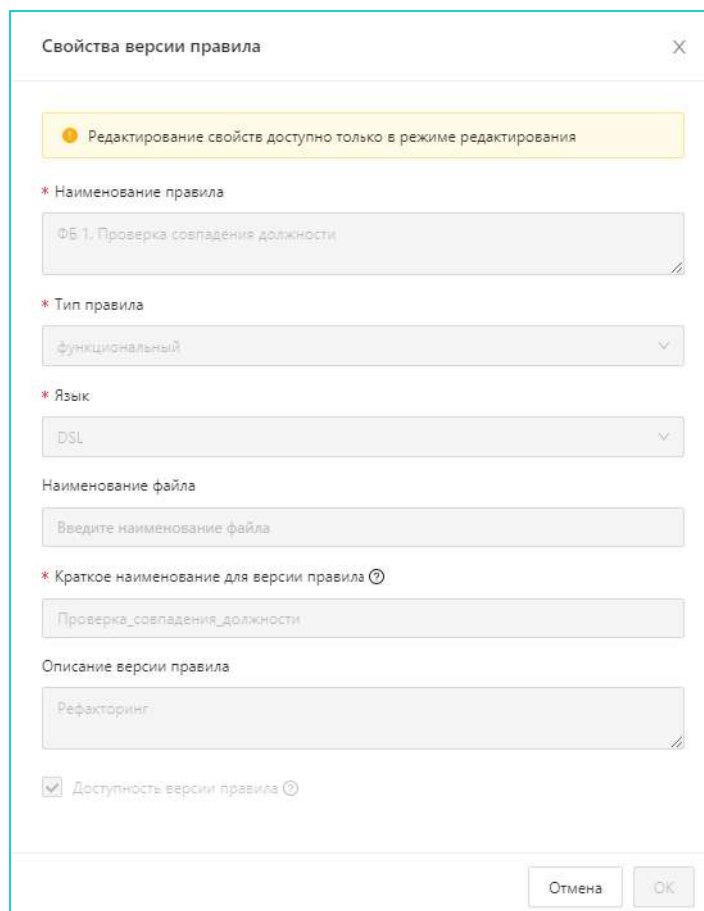


При удалении версии правила история публикаций сохраняется.

## Редактирование свойств правила

Панель редактирования позволяет редактировать следующие элементы:

- Наименование правила
- Тип правила
- Язык
- Наименование файла
- Краткое наименование
- Описание версии правила



The image shows a dialog box titled "Свойства версии правила" (Properties of rule version) with a close button (X) in the top right corner. Inside the dialog, there is a yellow warning box at the top stating: "Редактирование свойств доступно только в режиме редактирования" (Editing properties is available only in edit mode). Below this, there are several form fields, each preceded by a red asterisk indicating they are required:

- \* Наименование правила**: A text field containing "ФБ 1. Проверка совпадения должности".
- \* Тип правила**: A dropdown menu showing "функциональный" (functional).
- \* Язык**: A dropdown menu showing "DSL".
- Наименование файла**: A text field with the placeholder "Введите наименование файла" (Enter file name).
- \* Краткое наименование для версии правила**: A text field containing "Проверка\_совпадения\_должности".
- Описание версии правила**: A text area containing "Рефакторинг".

At the bottom of the form, there is a checkbox labeled "Доступность версии правила" (Availability of rule version) which is currently checked. In the bottom right corner of the dialog, there are two buttons: "Отмена" (Cancel) and "ОК" (OK).

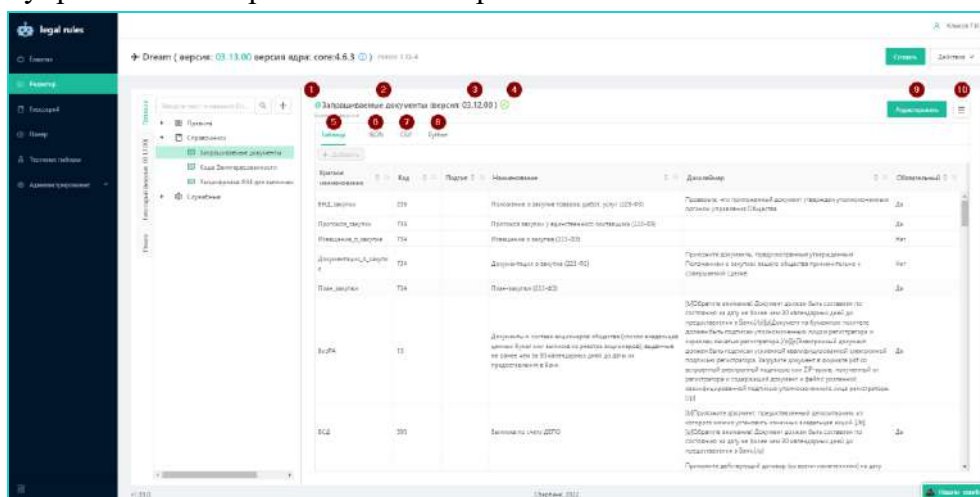
Также можно сделать правило недоступным для тестирования и сборки с помощью снятия соответствующей галочки.

## Панель для исполняемого блока типа «Справочники»

### Структура

Панель открывается при выборе исполняемого блока и содержит следующие элементы:

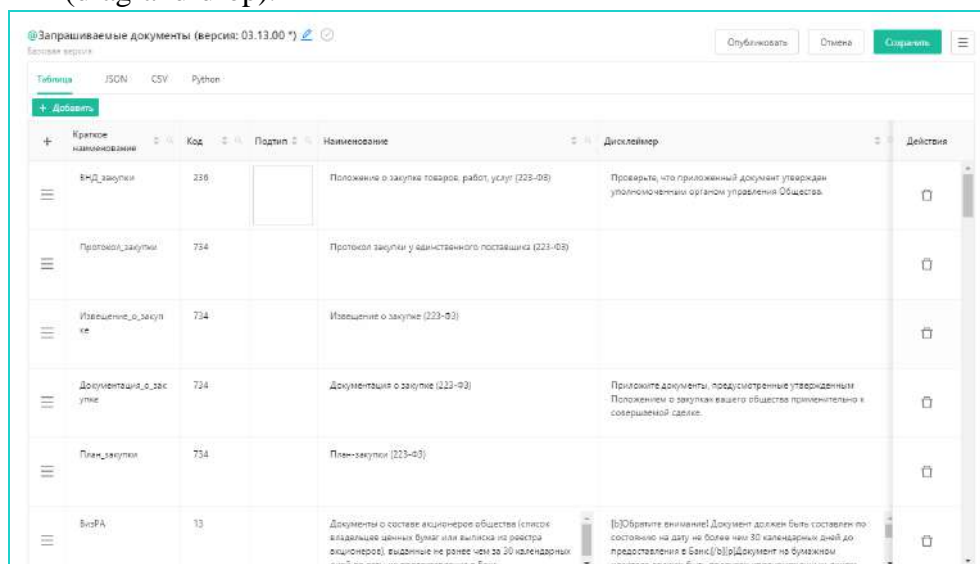
1. Метка справочника, при наведении отображается краткое наименование для его вызова;
2. Наименование блока: меняется в настройках блока и не связано с наименованием блока в дереве
3. Версия блока правил
4. Метка акцепта изменений в версии правил (Ревью)
5. Вкладка со справочником в табличном формате: столбцы таблицы позволяют производить поиск и сортировку данных
6. Вкладка со справочником в JSON-формате (режим просмотра в виде кода или дерева, возможность скачать JSON)
7. Вкладка со справочником в CSV-формате (возможность скачать файл в формате CSV)
8. Вкладка со справочником в Python-формате
9. Кнопка редактирования блока правил
10. Панель управления настройками блока правил



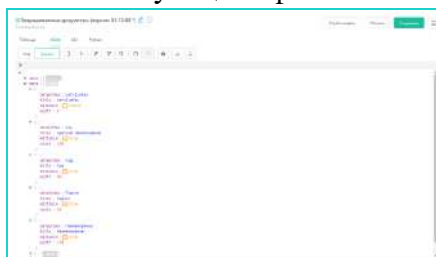
Основные принципы работы со справочником аналогичны исполняемым блокам «Правило» и «Функциональный блок». Ниже будет описан дополнительный функционал, характерный конкретно для данного типа исполняемого блока.

### Редактирование блока

Вкладка «Таблица» позволяет добавлять, редактировать и удалять строки таблицы, а также перетаскивать их (drag-and-drop).



Во вкладках «JSON», «CSV» и «Python» отображается информация из таблицы в соответствующем формате. Через вкладку CSV возможно создать таблицу через загрузку соответствующего файла.



## Управление настройками блока

Кнопка управления настройками аналогична блокам «Правило» и «Функциональный блок».

## Редактирование свойств правила

Панель редактирования позволяет редактировать следующие элементы:

- Наименование правила
- Краткое наименование
- Описание версии
- Блок добавления и редактирования колонок таблицы

Свойства версии правила

● Редактирование свойств доступно только в режиме редактирования

\* Наименование правила  
Запрашиваемые документы

\* Краткое наименование для версии правила ⓘ  
Запрашиваемые документы

Описание версии правила  
Базовая версия

☒ Доступность версии правила ⓘ

Колонки

| Название             | Ширина |
|----------------------|--------|
| Краткое наименование | 150    |
| Код                  | 90     |
| Подтип               | 90     |
| Наименование         | 370    |

Отмена OK

Также можно сделать правило недоступным для тестирования и сборки с помощью снятия соответствующей галочки.

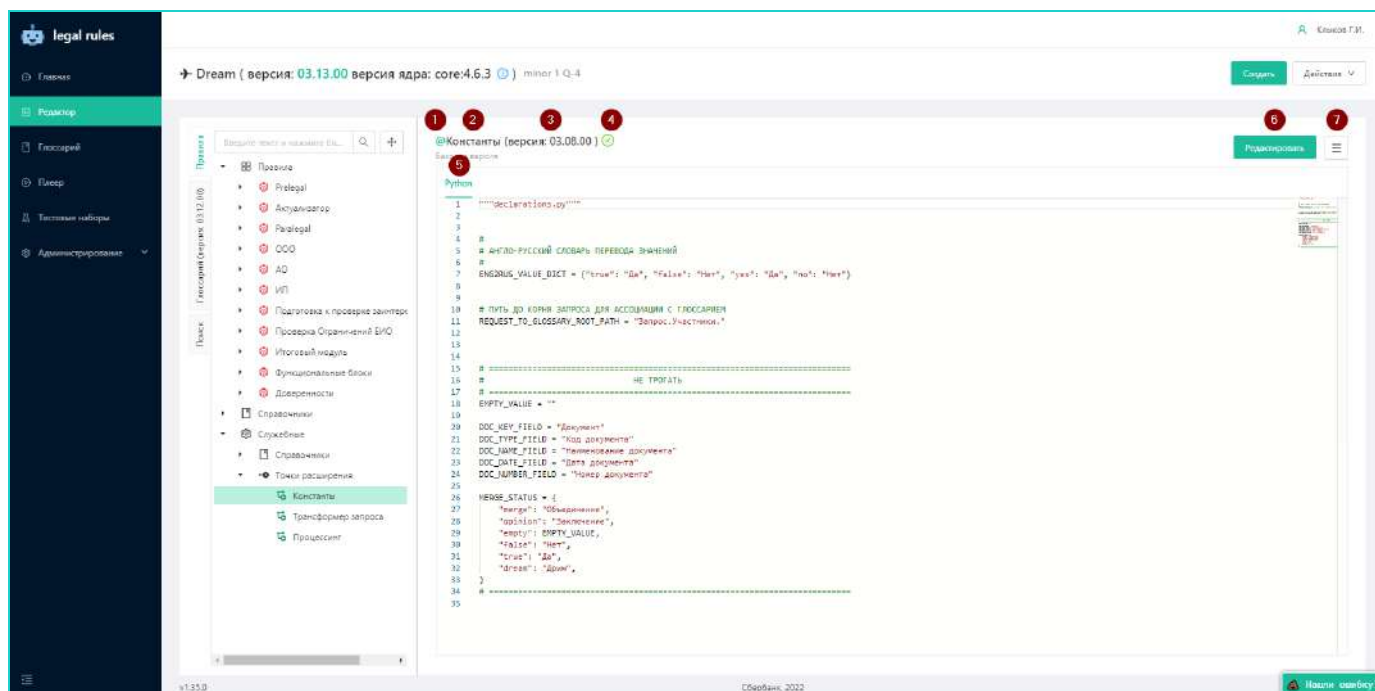
## Панель для исполняемого блока типа «Служебные - Справочники»

Работа с блоками подмодуля «Справочники» осуществляется аналогично работе с обычными справочниками.

## Панель для исполняемого блока типа «Служебные – Точки расширения» Структура

Панель блоков подмодуля «Точки расширения» открывается при выборе исполняемого блока и содержит следующие элементы:

1. Метка точки расширения, при наведении отображается краткое наименование для его вызова
2. Наименование блока: меняется в настройках блока и не связано с наименованием блока в дереве
3. Версия блока правил
4. Метка акцепта изменений в версии правил (Ревью)
5. Вкладка с кодом на языке Python
6. Кнопка редактирования блока правил
7. Панель управления настройками блока правил



Основные принципы работы со справочником аналогичны исполняемым блокам «Правило» и «Функциональный блок». Ниже будет описан дополнительный функционал, характерный конкретно для данного типа исполняемого блока.



## Редактирование свойств правила

Панель редактирования позволяет редактировать следующие элементы:

- Наименование правила
- Тип правила: «точка расширения»
- Язык: «Python»
- Наименование файла
- Краткое наименование
- Описание версии

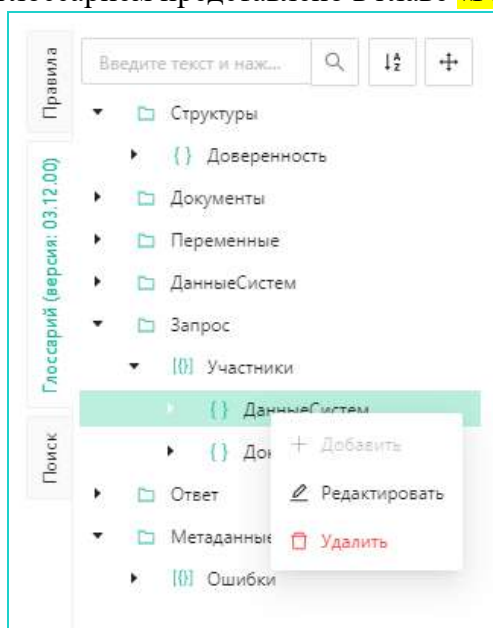
The screenshot shows a dialog box titled "Свойства версии правила" (Properties of rule version). It contains several input fields and a checkbox. At the top, a yellow banner states: "Редактирование свойств доступно только в режиме редактирования" (Editing properties is available only in edit mode). The fields are: "Наименование правила" (Rule name) with the value "Константы", "Тип правила" (Rule type) with a dropdown showing "точка расширения", "Язык" (Language) with a dropdown showing "Python", "Наименование файла" (File name) with the value "constants.py", "Краткое наименование для версии правила" (Short name for rule version) with the value "Константы", and "Описание версии правила" (Rule version description) with the value "Базовая версия". At the bottom, there is a checked checkbox labeled "Доступность версии правила" (Rule version availability) and two buttons: "Отмена" (Cancel) and "ОК" (OK).

Также можно сделать правило недоступным для тестирования и сборки с помощью снятия соответствующей галочки.

## Работа с вкладкой «Глоссарий»

Глоссарий представляет собой древовидную структуру, описывающую данные, используемые при написании кода. В данной вкладке отображается структура глоссария, привязанная к выбранной версии проекта.

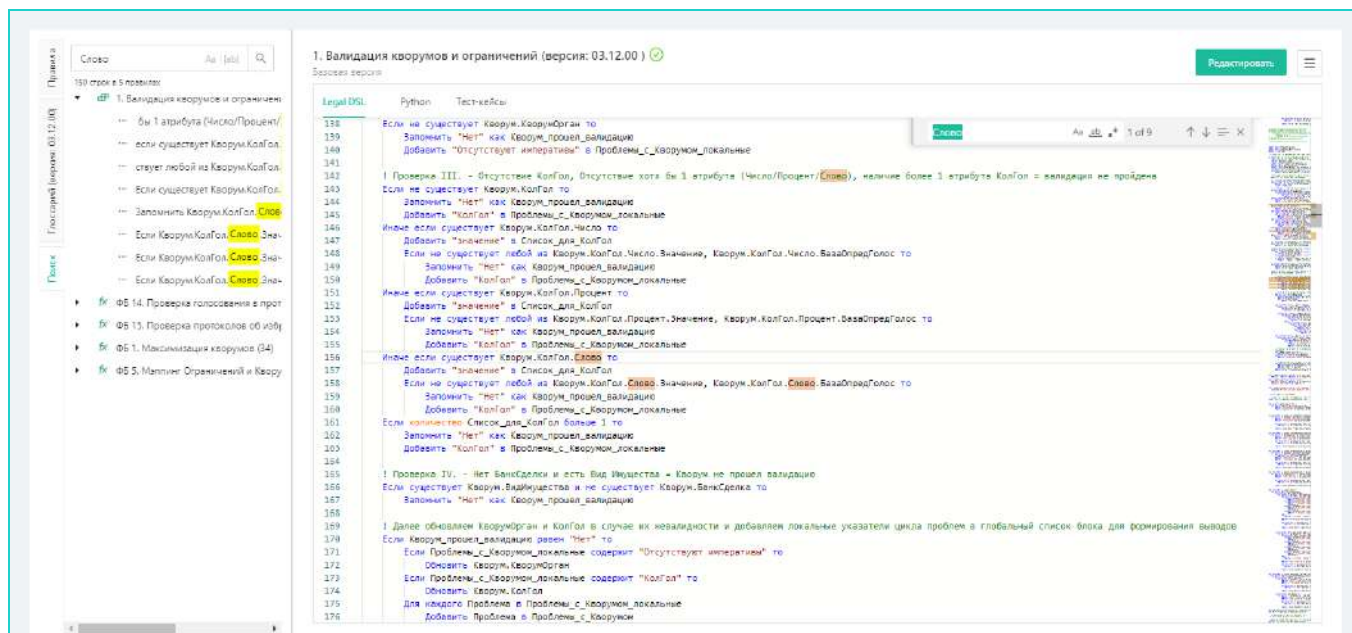
Подробное описание работы с глоссарием представлено в главе «Работа с вкладкой «Глоссарий»».



## Работа с вкладкой «Поиск»

Вкладка используется для поиска фрагментов текста по всем блокам правил в текущей версии проекта. Возможен поиск с учетом регистра или поиск слова целиком.

Результатом поиска является список блоков и строк, в которых было найдено совпадение. При нажатии на строку происходит переход в соответствующую позицию в блоке.



## Создание новой версии проекта

При нажатии кнопки «Создать» откроется окно, в котором необходимо указать:

- Родитель: версия проекта, которая будет взята за основу
- Новая версия: номер создаваемой версии проекта
- Описание: описание создаваемой версии проекта

При создании новой версии проекта подтягиваются версии глоссария и ядра, привязанные к родительской версии проекта.



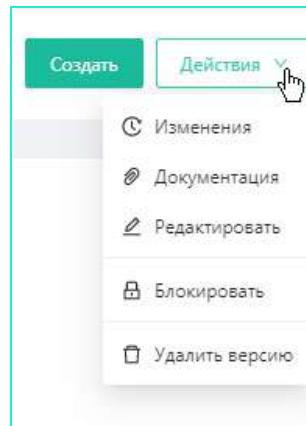
# Управление настройками текущей версии проекта

## Структура

При нажатии кнопки «Действия» откроется панель, содержащая следующие элементы:

- Изменения: версии правил, измененные в рамках версии проекта
- Документация: вложения для версии проекта
- Редактировать: изменение параметров версии проекта
- Блокировать: блокировка возможности изменения параметров проекта и вкладок «Правила» и «Глоссарий»
- Удалить версию: удаление текущей версии проекта

**! Не рекомендуется удалять версию проекта, если следующая версия содержит часть исполняемых блоков из удаляемой версии**



## Таблица правил, измененных в рамках версии проекта

При нажатии кнопки «Изменения» появляется таблица, которая содержит следующие столбцы:

- Правило: наименование измененного правила (при нажатии осуществляется переход к правилу)
- Версия: версия проекта, в рамках которого было изменено правило
- Ревью: была ли проведена проверка правила владельцем проекта (роль «BUSINESS\_ADMIN»)

Принимает следующие значения:

- Пустое поле: правило не проверено
- Review (✓): правило проверено
- Warning (⚠): после создания правила в текущей версии проекта, данное правило было изменено в предыдущей версии проекта. Требуется проверка, необходимо ли перенести изменения в текущую версию правил и проставление отметки review.
- История: переход в историю изменений правила
- Пользователь: логин пользователя, внесшего последнее изменение
- Дата: дата и время внесения последнего изменения

## Редактирование версии проекта

При нажатии кнопки «Редактировать» появляется окно, которое содержит следующие элементы:

- Версия: нумерация версии проекта (запрещено выбирать существующую версию)
- Описание: изменить описание версии проекта
- Версия РЮ: изменить версию ядра, привязанную к версии проекта

- Версия глоссария: изменить версию глоссария, привязанную к версии проекта
- URL адрес Jira: ссылка на пространство Jira
- Идентификаторы JiraID: идентификатор пространства Jira
- Схема запроса: JSON-схема входящего запроса (окно для загрузки появляется при проставлении галочки в соответствующем поле)
- Схема ответа: JSON-схема исходящего ответа (окно для загрузки появляется при проставлении галочки в соответствующем поле)

Редактирование версии проекта

Версия: 01.01.04

Описание: РЕФАК

Версия РЮ: core4.6.3

Версия глоссария: 02.00.00

URL адрес Jira: https://jira.cs.sbrf.ru/browse/

Идентификаторы JiraID: LDA

☒ Схема запроса: request\_scheme\_7.0.4.1.json

☒ Схема ответа: response\_schema\_2.3.json

Отмена OK

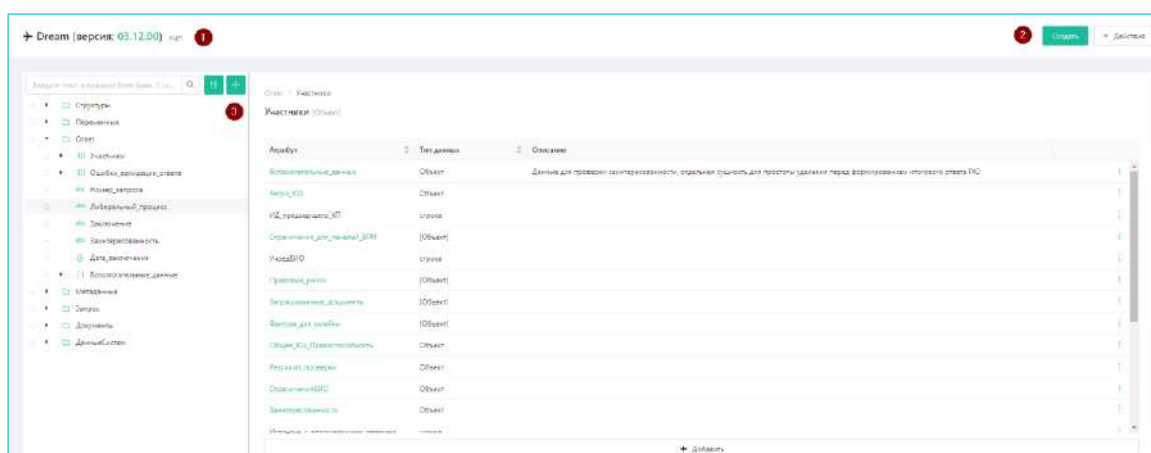
## Работа с вкладкой «Глоссарий»

### Структура страницы


Глоссарий представляет собой древовидную структуру, описывающую данные, используемые при написании кода.

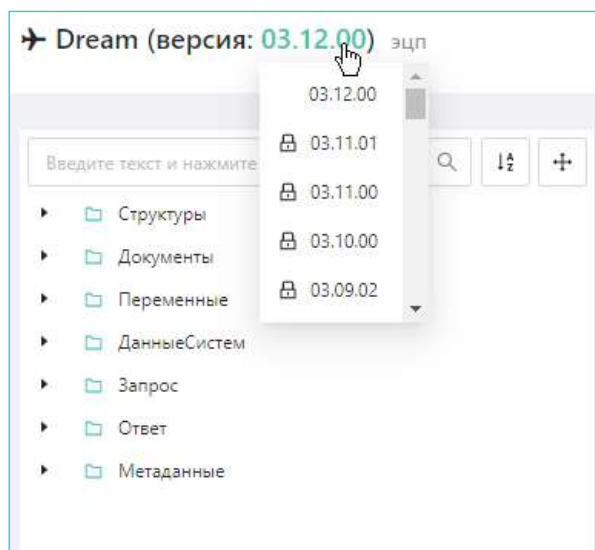
При переходе на вкладку «Глоссарий» отображается:

1. Версия и описание глоссария
2. Кнопки создания глоссария и управления его настройками
3. Структура глоссария и панель с описанием его элементов



## Версия глоссария

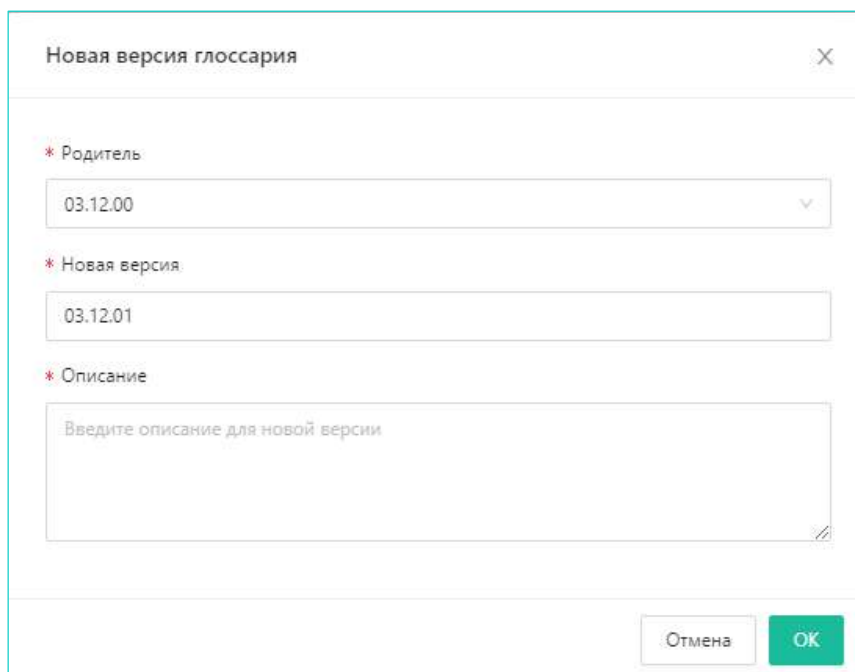
При наведении возможен выбор версии глоссария. Версии со значком  являются заблокированными и не доступны для редактирования.



## Создание новой версии глоссария

При нажатии кнопки «Создать» появляется окно, содержащее следующие элементы:

- Родитель: откуда наследуется структура глоссария в новую версию
- Новая версия: номер новой версии
- Описание: описание к новой версии глоссария

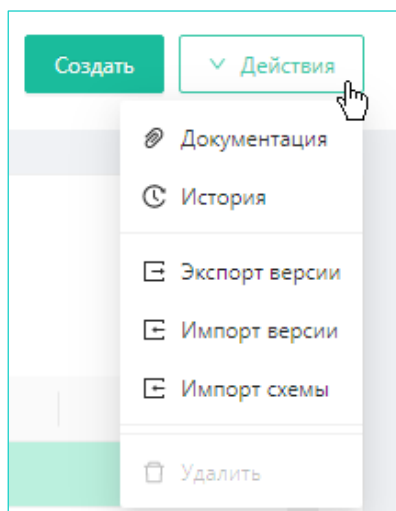
The screenshot shows a dialog box titled 'Новая версия глоссария'. It contains three fields: 'Родитель' (Parent) with a dropdown menu showing '03.12.00', 'Новая версия' (New version) with a text input field containing '03.12.01', and 'Описание' (Description) with a text area containing the placeholder 'Введите описание для новой версии'. At the bottom right, there are two buttons: 'Отмена' (Cancel) and 'ОК'.

# Управление настройками текущей версии глоссария

## Структура

При нажатии кнопки «Действия» откроется панель, содержащая следующие элементы:

- Документация: вложения для версии глоссария
- История: информация о внесенных изменениях в разрезе всех версий глоссария
- Экспорт версии: загрузка файла в формате CSV на локальный компьютер
- Импорт версии: построение глоссария из загружаемого файла в формате CSV
- Импорт схемы: загрузка интеграционных схем запроса/ответа
- Удалить: удаление версии глоссария



## История изменений глоссария

Вкладка «История» содержит информацию о внесенных изменениях в разрезе всех версий глоссария и состоит из следующих столбцов:

- Версия: версия глоссария, куда вносились изменения
- Пользователь: имя пользователя, внесившего изменения
- Дата: дата и время внесения изменения
- Сущность: элемент глоссария, который подвергся изменениям
- Событие: указывает каким образом были внесены правки («Создание», «Редактирование», «Перемещение», «Удаление», «Импорт», «Клонирование»)
- Путь: путь до измененного элемента
- Детали: подробная информация о сути изменения в формате JSON

| История + Export |               |                     |                   |                |   |           |
|------------------|---------------|---------------------|-------------------|----------------|---|-----------|
| Версия           | Пользователь  | Дата                | Сущность          | Событие        | Путь  | Детали    |
| 03.12.00         | kaisudkin1-mm | 01.09.2022 18:18:54 | Элемент глоссария | Создание       | Глоссарий / Ответ / Участники / Управляемое_Общество / Role                 | Подробнее |
| 03.12.00         | kaisudkin1-mm | 01.09.2022 18:18:40 | Элемент глоссария | Создание       | Глоссарий / Ответ / Участники / Управляемое_Общество / ID_участника         | Подробнее |
| 03.12.00         | kaisudkin1-mm | 01.09.2022 18:18:21 | Элемент глоссария | Создание       | Глоссарий / Ответ / Участники / Управляемое_Общество                        | Подробнее |
| 03.12.00         | kaisudkin1-mm | 31.08.2022 11:06:41 | Версия глоссария  | Импорт         |   | Подробнее |
| 03.12.00         | Kykov-SI      | 23.08.2022 11:01:46 | Элемент глоссария | Редактирование | Глоссарий / Переменные / Запрашиваемые_документы / Дисплейер                | Подробнее |
| 03.12.00         | Kykov-SI      | 23.08.2022 11:29:23 | Элемент глоссария | Редактирование | Глоссарий / Переменные / Запрашиваемые_документы / Дисплейер                | Подробнее |
| 03.12.00         | Troimov-I-S   | 16.08.2022 10:58:09 | Элемент глоссария | Создание       | Глоссарий / Ответ / Результат_проверки / Несоответствие_подписанта          | Подробнее |
| 03.12.00         | kaisudkin1-mm | 16.08.2022 10:30:32 | Элемент глоссария | Редактирование | Глоссарий / ДанныеСистем / Данные_по_закупкам / Реестр_организаций / 08_44  | Подробнее |
| 03.12.00         | kaisudkin1-mm | 16.08.2022 10:30:07 | Элемент глоссария | Редактирование | Глоссарий / ДанныеСистем / Данные_по_закупкам / Реестр_организаций / 08_223 | Подробнее |
| 03.11.01         | kaisudkin1-mm | 16.08.2022 10:22:10 | Элемент глоссария | Редактирование | Глоссарий / ДанныеСистем / Данные_по_закупкам / Реестр_организаций / 08_44  | Подробнее |
| 03.11.01         | kaisudkin1-mm | 16.08.2022 10:22:03 | Элемент глоссария | Редактирование | Глоссарий / ДанныеСистем / Данные_по_закупкам / Реестр_организаций / 08_223 | Подробнее |
| 03.11.01         | kaisudkin1-mm | 16.08.2022 10:15:08 | Версия глоссария  | Клонирование   |   | Подробнее |

## Экспорт версии глоссария

Команда производит загрузку файла с глоссарием на локальный компьютер в формате CSV. Файл имеет следующую структуру:

- Путь (обязательное поле): путь к элементу.  
Составляется путем соединения через точку без пробелов элементов глоссария от корня до конечного соответствующего элемента. Структура формируется итеративно – начиная с корня и далее наращивая элементы. Каждый новый элемент записывается ниже его корневой структуры.
- Тип данных (обязательное поле): определяет множество значений элемента и возможные операции над ним. Может принимать любое из следующих значений: «Объект», «строка», «число», «дата», «деньги», «период». Более подробное описание типов данных представлено ниже.
- Список (обязательное поле): является ли данный атрибут множественным (списочным). Если в поле выбран вариант «Да», тип данных обозначает тип данных всех элементов в списке.
- Структура - шаблонный тип данных, от которого наследована структура. Более подробное описание структуры представлено ниже.
- Описание – пользовательское описание атрибута.
- Свойства – иные технические данные элемента.

| Путь   | Тип данных | Список | Структура | Описание | Свойства   |
|--|------------|--------|-----------|----------|--|
| Документы  | Объект     | нет    |           |          |  |
| Документы_ДС_к_КоопДоговору  | Объект     | да     |           | Код: 605 | 0  |
| Документы_ДС_к_КоопДоговору_Идентификатор  | строка     | нет    |           |          | 0  |
| Документы_Свидетельство_67_1_ГК_РФ   | Объект     | да     |           | Код: 608 | "docDate", "ДатаПроведенияСоп", "docName", "Свидетельство_67_1_ГК_РФ", "docType" |
| Документы_Свидетельство_67_1_ГК_РФ_638_УО  | Объект     | да     |           |          |  |
| Документы_Свидетельство_67_1_ГК_РФ_638_УО_Голосование  | Объект     | нет    |           |          |  |
| Документы_Свидетельство_67_1_ГК_РФ_638_УО_Голосование_ГолосованиеЗа                            | Объект     | да     |           |          |  |
| Документы_Свидетельство_67_1_ГК_РФ_638_УО_Голосование_ГолосованиеЗа_КолГол                     | Объект     | да     |           |          |  |
| Документы_Свидетельство_67_1_ГК_РФ_638_УО_Голосование_ГолосованиеЗа_КолГол_Число               | денеги     | нет    |           |          |  |
| Документы_Свидетельство_67_1_ГК_РФ_638_УО_Голосование_ГолосованиеЗа_КолГол_Промент             | денеги     | нет    |           |          |  |
| Документы_Свидетельство_67_1_ГК_РФ_638_УО_Голосование_ГолосованиеЗа_КолГол_ВзаимПротивОпас     | строка     | нет    |           |          |  |
| Документы_Свидетельство_67_1_ГК_РФ_638_УО_Голосование_ГолосованиеЗа_Голосование                | Объект     | да     |           |          |  |
| Документы_Свидетельство_67_1_ГК_РФ_638_УО_Голосование_ГолосованиеЗа_Голосование_ВУО            | строка     | да     |           |          |  |
| Документы_Свидетельство_67_1_ГК_РФ_638_УО_Голосование_ГолосованиеЗа_Голосование_РешОП          | строка     | да     |           |          |  |
| Документы_Свидетельство_67_1_ГК_РФ_638_УО_Голосование_ГолосованиеЗа_Голосование_ИтоОП          | строка     | да     |           |          |  |
| Документы_Свидетельство_67_1_ГК_РФ_638_УО_Голосование_ГолосованиеЗа_Голосование_СМО            | строка     | нет    |           |          |  |
| Документы_Свидетельство_67_1_ГК_РФ_638_УО_Голосование_ГолосованиеЗа_Голосование_ПНО            | строка     | нет    |           |          |  |
| Документы_Свидетельство_67_1_ГК_РФ_638_УО_Голосование_ГолосованиеПротив                        | Объект     | да     |           |          |  |
| Документы_Свидетельство_67_1_ГК_РФ_638_УО_Голосование_ГолосованиеПротив_КолГол                 | Объект     | да     |           |          |  |
| Документы_Свидетельство_67_1_ГК_РФ_638_УО_Голосование_ГолосованиеПротив_КолГол_Число           | денеги     | нет    |           |          |  |
| Документы_Свидетельство_67_1_ГК_РФ_638_УО_Голосование_ГолосованиеПротив_КолГол_Промент         | денеги     | нет    |           |          |  |
| Документы_Свидетельство_67_1_ГК_РФ_638_УО_Голосование_ГолосованиеПротив_КолГол_ВзаимПротивОпас | строка     | нет    |           |          |  |

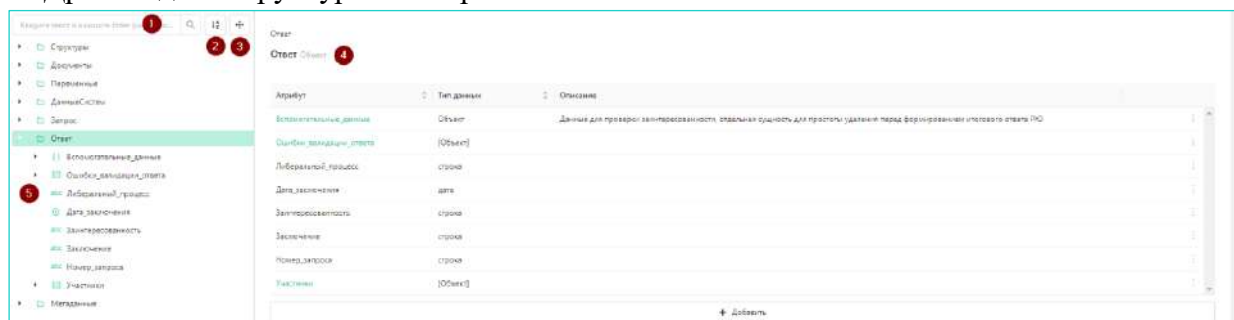
**! При внесении изменений не должна нарушаться иерархическая структура глоссария, а также должны быть заполнены все обязательные поля. В противном случае импорт файла и построение глоссария будут невозможны.**

## Работа со структурой глоссария и панелью его элементов

### Описание инструментов

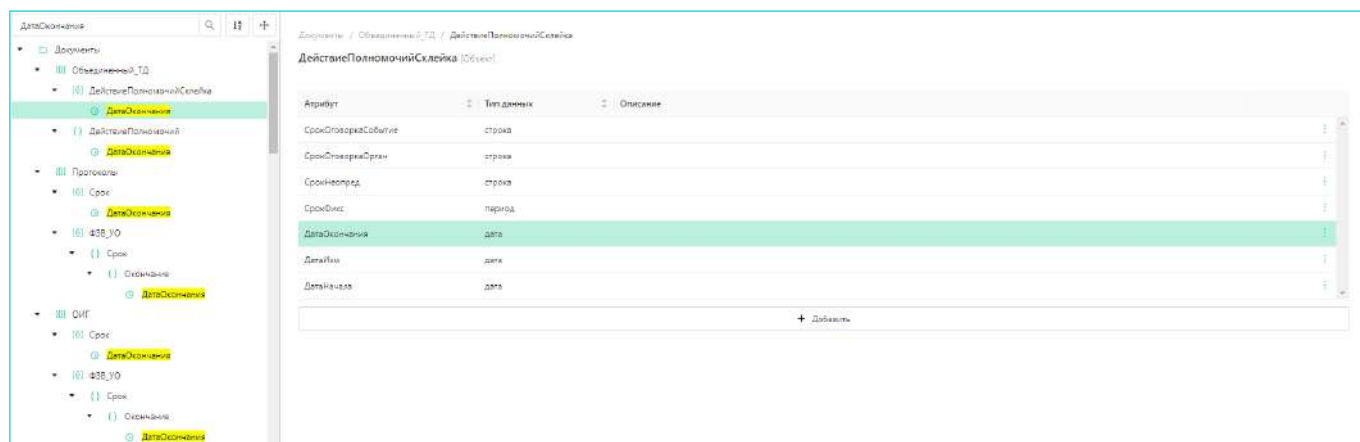
В данной области содержатся следующие поля:

1. Поиск элементов в структуре глоссария;
2. Сортировка элементов в алфавитном порядке (при повторном нажатии сортирует в обратном алфавитном порядке);
3. Включение возможности перемещения атрибутов по структуре глоссария (drag-and-drop);
4. Панель с описанием структуры;
5. Древоподобная структура глоссария.



## Поиск атрибутов в структуре глоссария

Результатом поиска является отфильтрованный глоссарий, в котором оставлены только искомые элементы с указанием пути до них. Возможен поиск с учетом регистра или поиск слова целиком.



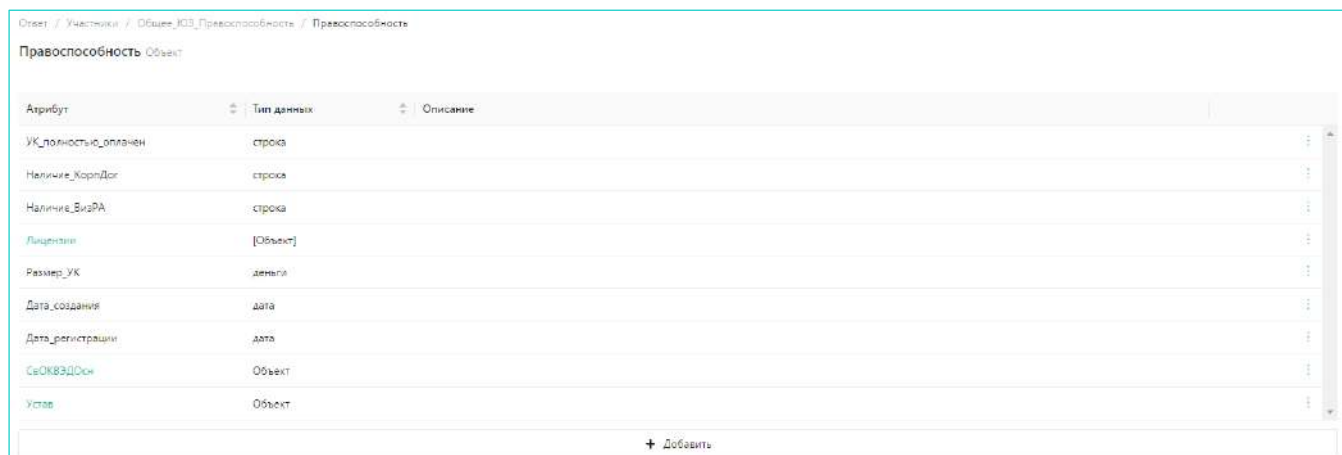
| Атрибут           | Тип данных | Описание |
|-------------------|------------|----------|
| СрокСпоркаСобытие | строка     |          |
| СрокСпоркаОрган   | строка     |          |
| СрокСопред        | строка     |          |
| СрокВнес          | период     |          |
| ДатаОсвоения      | дата       |          |
| ДатаПлю           | дата       |          |
| ДатаНачала        | дата       |          |

## Визуализация структуры глоссария

При выборе элемента в дереве отображает подробную информацию об элементах:

- Если элемент имеет тип «Объект» или «[Объект]» (список объектов), отображаются все элементы находящиеся в нем
- В остальных случаях отображаются все элементы, находящиеся в том же корне, что и выбранный

Возможен переход в структуру элемента с типами данных «Объект» или «[Объект]» по нажатию на них. Данные элементы выделены **цветом**.



| Атрибут              | Тип данных | Описание |
|----------------------|------------|----------|
| УК_полностью_оплачен | строка     |          |
| Наличие_ХорнДог      | строка     |          |
| Наличие_ВизРА        | строка     |          |
| Лицензии             | [Объект]   |          |
| Размер_УК            | деньги     |          |
| Дата_создания        | дата       |          |
| Дата_регистрации     | дата       |          |
| СвОК83ЦОсн           | Объект     |          |
| Устав                | Объект     |          |

## Древовидная структура глоссария

Отображает древовидную иерархическую структуру глоссария.

Корневые элементы ({}):

- Структуры: содержит шаблонные структуры, которые могут наследоваться в других элементах, для исключения дублирования
- Переменные - глобальные переменные
- Запрос: структура запроса после предобработки на стороне webDSL
- Ответ - структура ответа, формирующегося по результатам работы

Типы данных:

- Объект ({}): содержит в себе вложенную структуру
- Строка (abc) – содержит в себе строковые значения
- Число (123) – содержит в себе числовые значения
- Дата (📅) – содержит в себе значения в форме даты в формате «ДД.ММ.ГГГГ» или «ГГГГ-ММ-ДД»
- Деньги (\$) – содержит в себе значения в денежном формате
- Период (📅) – содержит в себе значения в форме периода в формате Г\_М\_Н\_Д  
Пример: «1 год 2 месяца 3 недели 4 дня» = «1\_2\_3\_4».

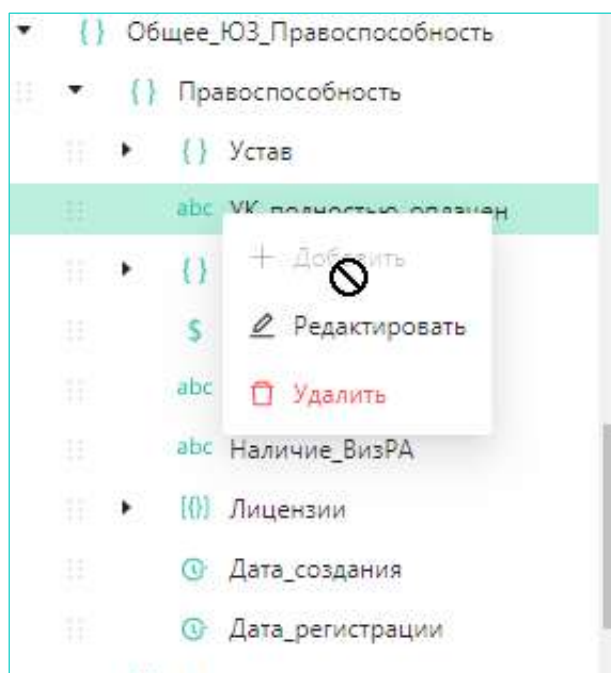
Любой из типов данных может быть единичный или списочный (отображается путем добавления с двух сторон квадратных скобок «[значок типа данных]», например «[abc]» = список строк).

## Структуры

Корень «Структуры» содержит шаблонные структуры, которые могут быть переиспользованы. Например, если несколько элементов глоссария содержат одинаковый набор атрибутов, то для исключения дублирования, создается структура, которая указывается в свойствах соответствующих элементов.

## Функционал по работе добавлению, редактированию и удалению атрибутов

Данный функционал доступен в дереве (клик правой кнопкой мыши на элементе), и в панели структуре глоссария (клик левой кнопкой мыши на соответствующем значке):



| Атрибут              | Тип данных | Описание |
|----------------------|------------|----------|
| УК_полностью_оплачен | строка     |          |
| Наличие_КорпДог      | строка     |          |
| Наличие_ВизРА        | строка     |          |
| Лицензии             | [Объект]   |          |
| Размер_УК            | деньги     |          |
| Дата_создания        | дата       |          |
| Дата_регистрации     | дата       |          |
| СвОКВЭДОск           | Объект     |          |
| Устав                | Объект     |          |

[Редактировать](#)  
[Удалить](#)

+ Добавить

При добавлении элемента необходимо указать следующие параметры:

- Наименование
- Описание
- Список: по умолчанию атрибут единичный, если поставлена галочка, то атрибут будет считаться множественным (списочным)
- Тип данных: если элемент отмечен, как списочный, тип данных обозначает тип данных всех элементов в списке
- Структура - шаблонный тип данных, от которого наследована структура

Новый элемент

Основные атрибуты

Доп. атрибуты

\* Наименование

Введите наименование элемента

Описание

Введите описание

☐ Список

\* Тип данных

Выберите тип данных

Структура

Не определено

Отмена

ОК

Вкладка «Доп. атрибуты» используется только для корня «Документы» (специфический корень для проекта K7M) и содержит следующие элементы:

- Код документа: код типа документа
- Наименование документа: отображаемое наименование документа для юриста
- Дата документа: атрибут документа, откуда берется дата для отображения юристу
- Номер документа: атрибут документа, откуда берется номер для отображения юристу



Редактирование элемента

Основные атрибуты

Доп. атрибуты

Код документа

Атрибут глоссария, содержащий код документа

Наименование документа

Атрибут глоссария, содержащий наименование документа

Дата документа

Атрибут глоссария, содержащий дату документа

Номер документа

Атрибут глоссария, содержащий номер документа

Отмена

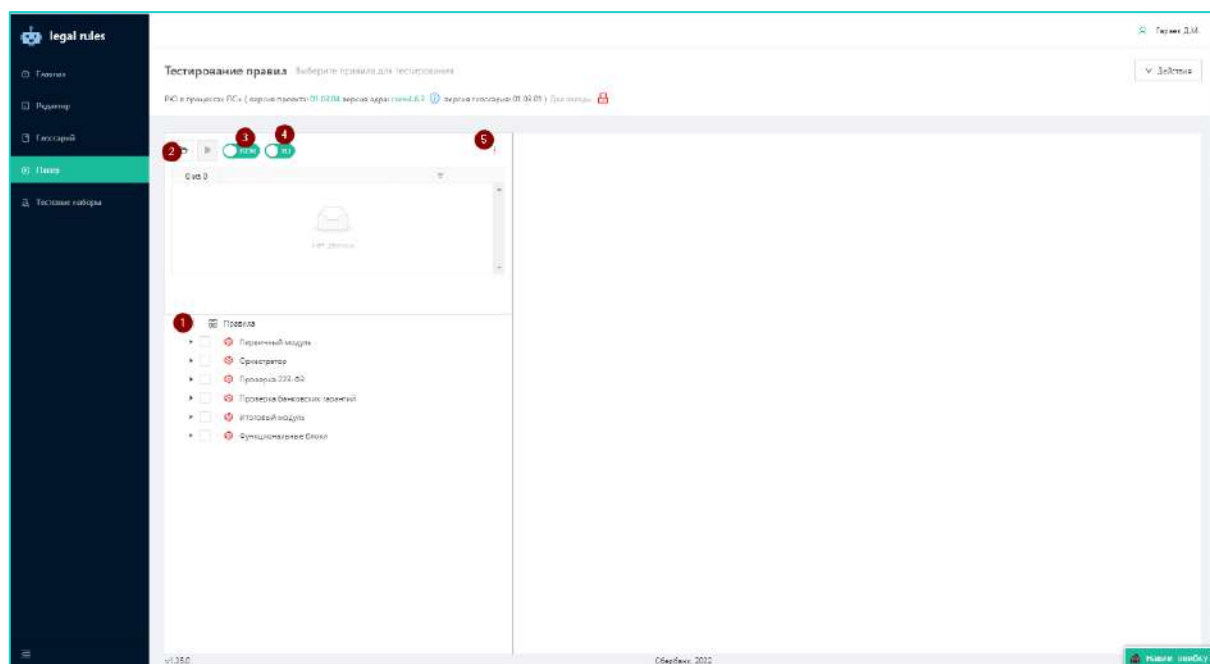
OK

## Работа с вкладкой «Плеер»

### Описание

Данная вкладка представляет собой инструмент тестирования правил и содержит следующие элементы:

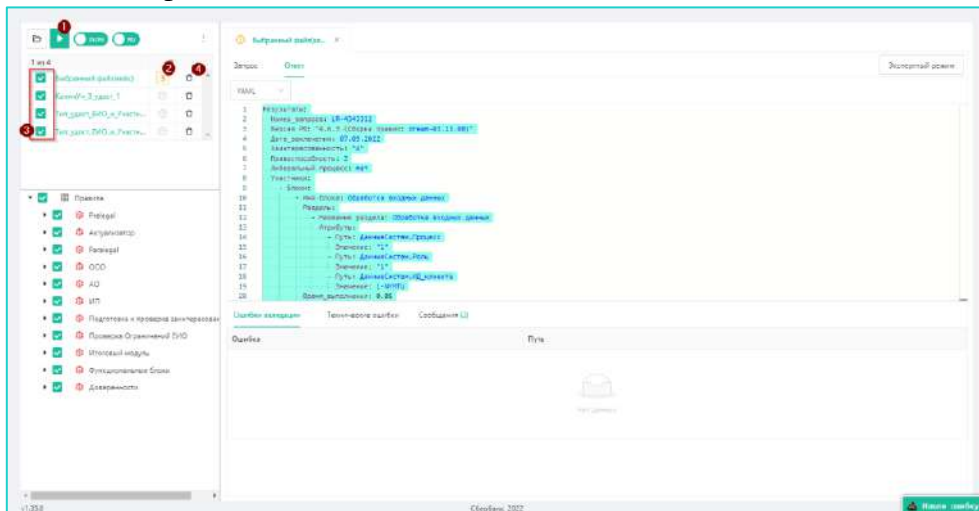
1. Выбор модулей для тестирования  
Переход между модулями настраивается пользователем с помощью оператора «Перейти в модуль». Переход между правилами внутри модуля осуществляется автоматически, возможно настроить принудительный переход (в рамках модуля) с помощью оператора «Перейти в блок».
2. Выбор файла для тестирования (тест-кейс в формате JSON)
3. Формат ответа
4. Язык ответа
5. Опции для работы со списком файлов



## Панель работы с Плеером

После выбора модулей для тестирования и выбора файла становятся доступны следующие функции:

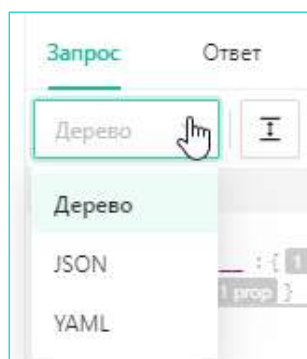
1. Запуск тестирования
2. Результат теста
3. Выбор файлов для тестирования (по умолчанию тестируются все файлы)
4. Удаление одного файла














## Работа с запросом (выбранным файлом)

Двойным кликом по названию файла, открывается функционал работы с запросом. Он представляет собой древовидную структуру:


1. Выбор вида просмотра запроса:
  - Дерево;
  - JSON;
  - YAML.



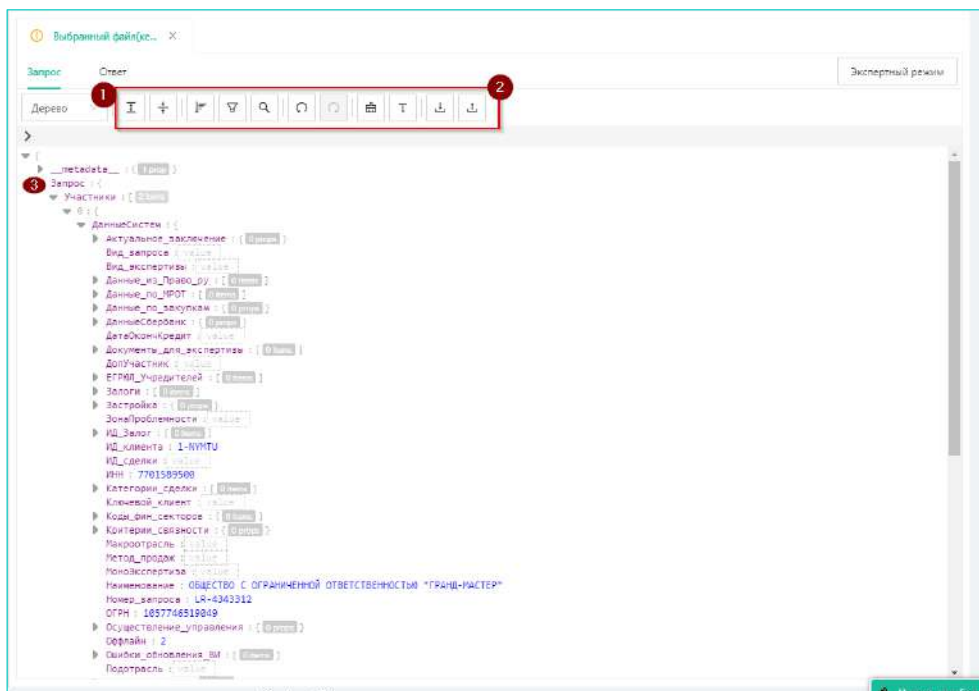
2. Панель инструментов для работы с запросом:


- Растянуть (  )
- Сжать (  )
- Сортировать (  )
- Преобразование (  )
- Поиск (  )
- Отменить (  )
- Вернуть (  )
- Удалить пустые ноды (  )
- Трансформировать JSON (  )
- Скачать JSON (  )
- Загрузка JSON (  )

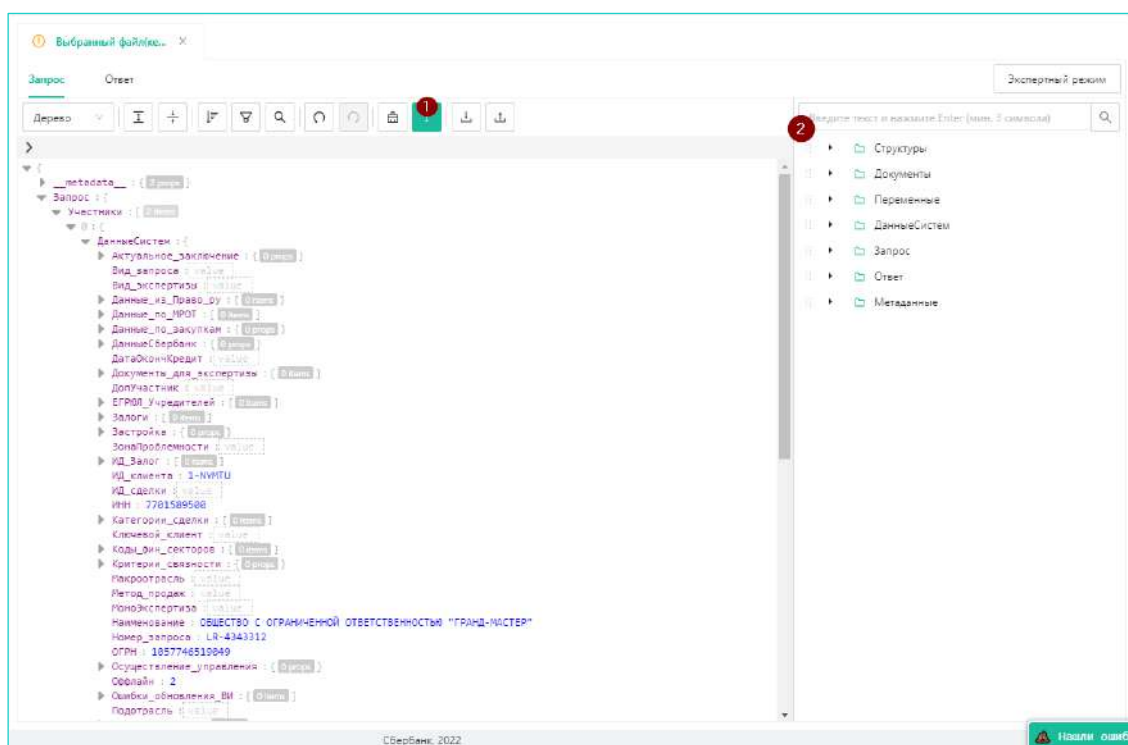
### 3. Взаимодействие со структурой запроса.

При нажатии на (  ) раскрывается запрос где возможно:

- Добавить
- Изменить
- Удалить



Перед началом работы с файлом, его необходимо трансформировать  (1), после трансформации появляется глоссарий (2).



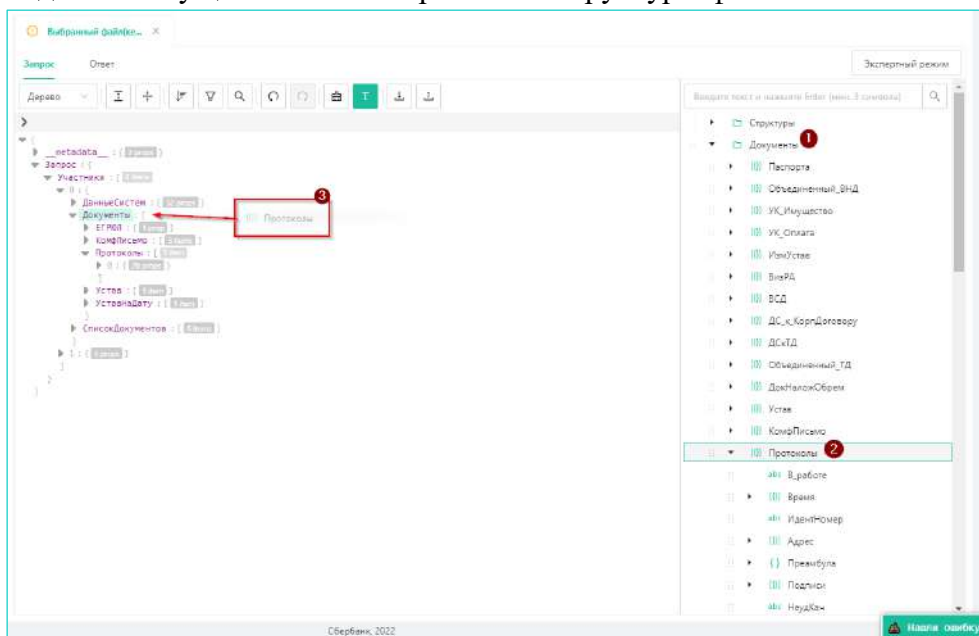
Появляется расширенная возможность для редактирования файла, позволяющая добавлять, удалять, изменять необходимые данные.

## Способы редактирования данных.

Возможен перенос необходимых элементов глоссария из вкладки, которая открывается при трансформации запроса (файла).


1. Открываем древовидную структура (в примере документы)
2. Находим необходимый элемент и переносим

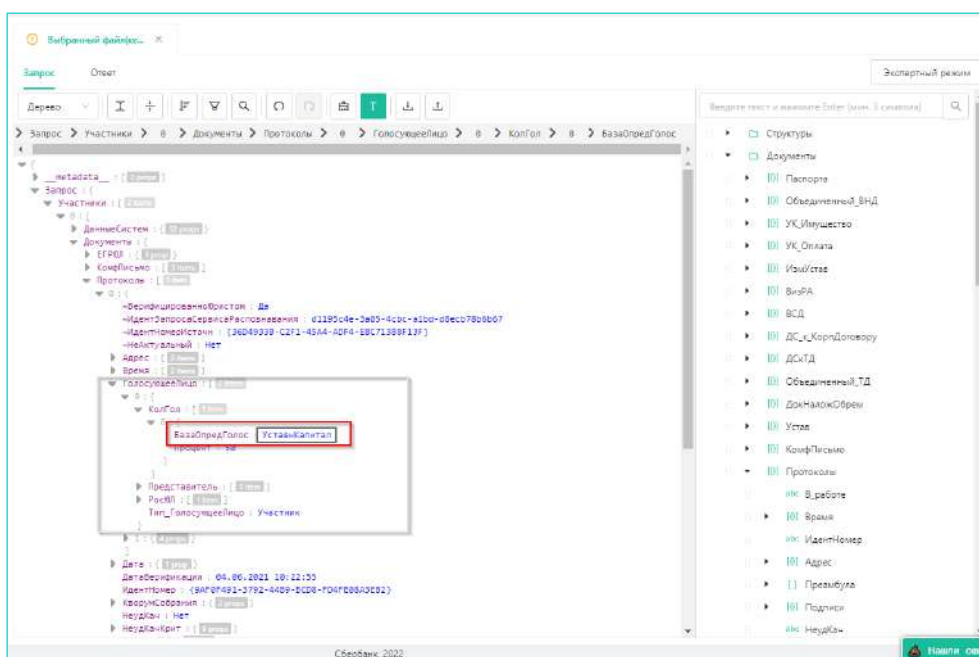
Перенос должен осуществляться сохранением структуры файла



## Изменение запроса

Изменение запроса (файла) происходит в древовидной структуре.

1. Раскрываем элемент, в который нужно внести изменения
2. Двойным нажатием переходим в режим редактирования значения и вносим изменения
3. После изменения нужно сохранить файл (скачать JSON )
4. Далее запускаем тест запроса

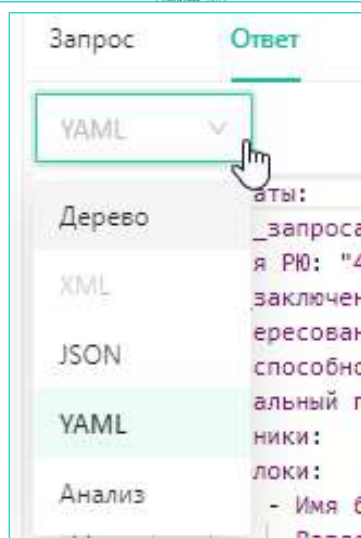
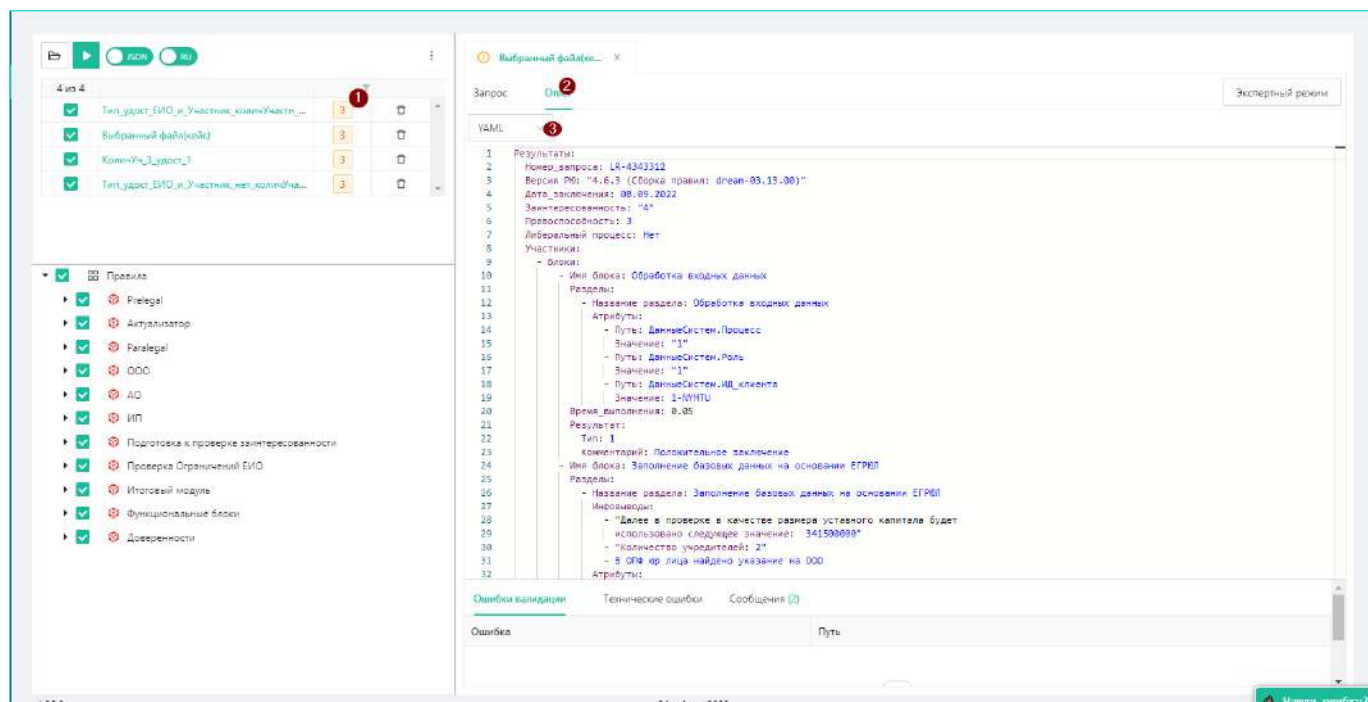


## Работа с ответом

После запуска тестирования появится результат обработки теста, содержащий следующие элементы:

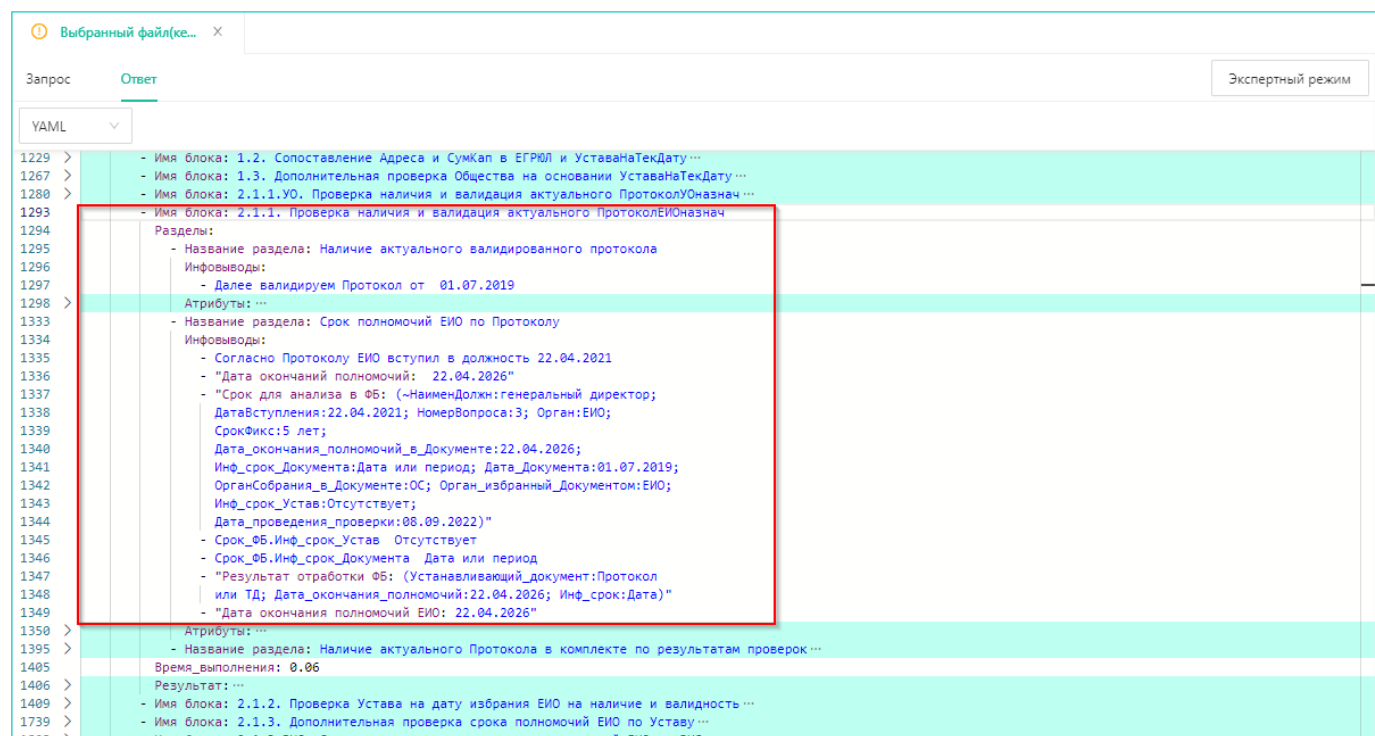
1. Результат теста: двойным кликом по названию файла, открывается функционал работы с запросом
2. Переход на вкладку «Ответ»
3. Выбор вида просмотра ответа

В профиле пользовательских настроек можно выбрать формат открытия ответа по умолчанию.



## Анализ ответа

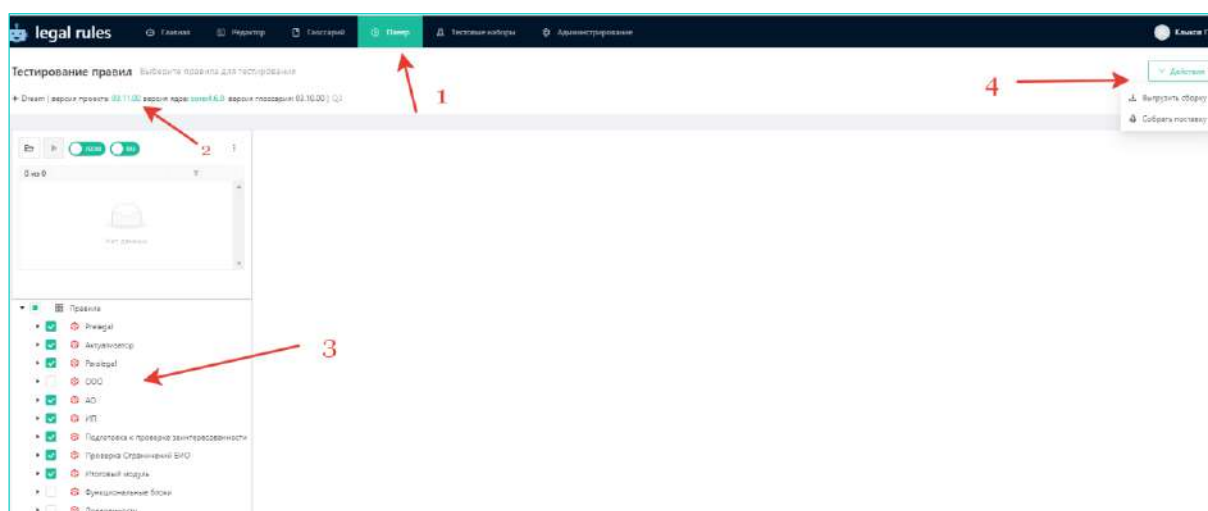
В ответе отображается результат отработки правил по каждому модулю/каталогу/правилу. Наименование блоков, разделов совпадает с указанными в коде. В результате можно увидеть все выводы и сообщения для проверки тестового сценария.



## Сборка поставки РЮ<sup>2</sup>

Для подготовки сборки правил РЮ к выгрузке необходимо:

1. Выбрать вкладку «Плеер»
2. Выбрать версию проекта, которую необходимо собрать
3. Выбрать необходимые модули правил
4. Выбрать требуемое действие



<sup>2</sup> Описанные действия доступны только для администратора.

При выгрузке сборки будет выведено предупреждение о наличии черновиков и неподтвержденных правил («Ревью») в выбранных модулях, а также название сборки. После нажатия кнопки «Запустить» на локальный компьютер будет загружен архив с версиями правил.

Выгрузка сборки

!

Сборка содержит черновики  
03.13.00 (1 шт) Будут выгружены только опубликованные версии правил

!

Сборка содержит неподтвержденные версии правил  
03.13.00 (10 шт), 03.12.00 (3 шт)

\* Номер сборки

dream- 03.13.00

-4.6.4-20220909165951

Отмена

Запустить

При выгрузке сборки будет выведено предупреждение о наличии черновиков и неподтвержденных правил ("Ревью") в выбранных модулях, а также название сборки и поле для ее описания. При наличии черновиков публикация сборки будет невозможна. После нажатия кнопки "Запустить" сборка будет опрарлена на стенд ИФТ.

Сборка поставки

✗

Сборка содержит черновики  
Публикация сборки запрещена!

!

Сборка содержит неподтвержденные версии правил  
03.13.00 (10 шт), 03.12.00 (3 шт)

\* Номер сборки

dream- 03.13.00

-4.6.4-20220909170057

\* Описание сборки

Введите описание сборки

Отмена

Запустить

## Работа с DSL

DSL – язык программирования с динамической строгой типизацией, ориентированный на повышение производительности разработчика, читаемости кода и его качества. В DSL выделяются блоки кода пробельными отступами, синтаксис минималистичен и интуитивно понятен.

## Базовый синтаксис

### Типы данных

Поддерживаются следующие типы данных:

- дата,
- число,
- деньги,
- период,
- списки,
- строки (по умолчанию).

Основными типами данных являются числа и строки. Числа записываются последовательностью цифр, для дробных чисел в качестве разделителя может использоваться «.» и «,». Строки записываются в кавычках. Таким образом 1 и «1» - это разные объекты, первый объект – число, а второй – строка. Операция «+» для чисел и строк работает по-разному: для чисел это сложение, для строк – конкатенация.

Дата

Тип поддерживает следующие нотации:

"19.02.12"

"19 февраля 2012"

"19 февраля 2012 года"

"2012-02-19"

Для данного типа доступны все основные математические и сравнительные операции, при этом:

Вычитание из одной даты другой даст на выходе переменную типа Период

Доступны операции сложения и вычитания даты с периодом, в результате будет получена новая переменная типа Дата

| Операции: | Примеры:  |
|-----------|---|
| Сложение  | Возможность реализации операций типа:<br><br>10.03.2018 + 1 месяц = 10.04.2018<br>10.03.2018 + 1 день = 11.03.2018<br>10.03.2018 + 1 год = 10.03.2019 |
| Вычитание | Возможность реализации операций типа:<br><br>31.07.2018 - 1 месяц = 30.06.2018<br><i>если даты нет, то берем ближайшую существующую</i>               |
| Сравнение | Возможность реализации операций типа:<br><br>Если Дата1 позже Дата2 то ...  |



|                     |  |
|---------------------|--|
|                     | Если Дата1 позже или равно Дата2 то ...<br>Если Дата1 раньше Дата2 то ...<br>Если Дата1 раньше или равно Дата2 то ...  |
| Нахождение периодов | Возможность реализации операций типа:<br><br>Если период от 01.01.01 до 05.02.01 меньше 1 года то ...<br>Если период от 01.01.01 до 05.02.01 больше 1 месяца то ...<br>Если период от 01.01.01 до 05.02.01 больше 20 дней то ... |

## Период

Тип хранит данные в структуре (Годы/Месяцы/Дни). При этом для операций сравнения и при присваивании значений данный тип также поддерживает элемент "Неделя", который при присвоении трансформируется в элемент "Дни".

При этом сравнение выполняется по следующим правилам:

- при сравнении двух периодов, в которых не заполнен элемент День, поэлементно сравниваются контейнеры Год и Месяц
- при сравнении двух периодов, в которых заполнен элемент День, значения всех элементов приводятся к дням, при этом учитывается високосный период и количество дней в месяцах по формуле:  $(365*3+366) / 4 / 12$

Тип поддерживает следующие нотации:

12\_3\_4\_12

5 мес.

1 год и 13 месяцев

10 лет 2 месяца 22 дня

## Число

Тип поддерживает все математические операции, при этом внутри себя поддерживается типизация как целых чисел (Integer), так и дробных (Decimal).

По умолчанию все значения переводятся в Decimal, при этом, если значение переменной целое, оно неявно приводится в дробное:

Пользователь пишет Запомнить 1 как Цифра типа Число - будет 1,00

Пользователь пишет Запомнить 1,25 как Цифра типа Число - будет 1,25

## Деньги

Тип является производным от дробного Decimal и реализован как самостоятельный класс с поддержкой широкого набора нотаций.

Поддержка валют: только валюты в номиналах рубли/копейки

## Список

Тип данных способный содержать в себе переменные различных типов. В текущей версии языка поддерживаются операции:

- добавление элемента
- сравнение списков
- фильтрация списков
- удаление элемента

- итерация по элементам списка

## Хранение данных

DSL хранит данные в Переменных, Списках и Словарях.

### Переменная

В языке используются переменные двух типов:

**Локальные** - к ним относятся все переменные, созданные в конкретном блоке правил и видимые только в пределах этого блока (такие переменные создаются с помощью команды **Запомнить**).

После выхода из блока, информация о значениях этих переменных утрачивается, сами переменные удаляются из памяти. Информацию из локальной переменной можно передать в глобальную, для возможности дальнейшего обращения к этому значению.

Пример: Запомнить «значение» как ЛокальнаяПеременная

После этого при обращении к ЛокальнаяПеременная будет получен результат «значение».

**Глобальные** - к ним относятся переменные, которые заложены в модель данных на уровне глоссария (то есть эти переменные предопределены заранее). Такие переменные создаются с помощью команды **Записать**.

В названии Переменные всех типов нельзя пробельные символы.

### Список

Используется для хранения структурированной информации. Элементы в список помещаются обезличено. Каждому элементу Списка присваивается индекс, однако отсутствует уникальный адрес, позволяющий обратиться к конкретному элементу. Можно хранить неограниченное и заранее неопределенное количество элементов.

### Словарь

Используется для хранения структурированной информации. Всем переменным необходимо присвоить уникальный адрес. Элементам в словаре не присваивается индекс, из-за чего невозможно проведение операций над всеми элементами сразу или использовать итерационное обращение.

## Ветвление

### Условные инструкции

Условные инструкции в DSL имеет следующий синтаксис:

Если Условие то

    Блок\_инструкций\_1

Иначе

    Блок\_инструкций\_2

Блок\_инструкций\_1 будет выполнен, если Условие истинно. Если Условие ложно, будет выполнен Блок\_инструкций\_2.

В условной инструкции может отсутствовать слово Иначе и последующий блок. Такая инструкция называется неполным ветвлением. Например, если дано число X и мы хотим заменить его на абсолютную величину x, то это можно сделать следующим образом:

Если  $X < 0$  то

$X = -X$

В этом примере переменной  $X$  будет присвоено значение  $-X$ , но только в том случае, когда  $X < 0$ . Для выделения блока инструкций, относящихся к инструкции Если или Иначе, в языке DSL используются отступы. Все инструкции, которые относятся к одному блоку, должны иметь равную величину отступа, то есть одинаковое число пробелов в начале строки. Необходимо использовать отступ, кратный четырем пробелам.

### Вложенные условные инструкции

Внутри условных инструкций можно использовать любые инструкции языка DSL, в том числе и условную инструкцию. Вложенное ветвление — после одной развилки в ходе исполнения программы появляется другая развилка. При этом вложенные блоки имеют на 4 пробела больший размер отступа, чем блок, в который они вложены.

## Комментарии

Для отметки, что делает тот или иной участок кода, применяются комментарии. При трансляции и выполнении программы интерпретатор игнорирует комментарии, поэтому они не оказывают никакого влияния на работу программы.

Комментарии предваряются восклицательным знаком «!». Они могут располагаться на отдельной строке.

Любой набор символов после знака «!» представляет комментарий.

Также они могут располагаться на той же строке, что и инструкции языка, после выполняемых инструкций.

## Блоки кода

### Функциональные блоки

Функциональные блоки кода отличаются от Основных блоков тем, что при выполнении программы интерпретатором линейно они не вызываются. Функциональные блоки могут вызываться из Основных блоков с последующим возвратом в Основной блок неограниченное количество раз и в любом порядке.

Функциональные блоки представляют блок кода, который выполняет определенную задачу и который можно повторно использовать в других частях программы.

Для выполнения Функционального блока в него подаются входные данные, по результатам его работы в Основной блок возвращаются выходные данные.

Пример:

В основном блоке:

Запомнить результат выполнения ИмяФункциональногоБлока для ВходныеДанные как ВыходныеДанные

В функциональном блоке:

Вернуть "Значение"

В результате выполнения Функционального блока в Основном блоке появится локальная переменная со значением «Значение».

Из Функционального блока можно вызывать другие Функциональные блоки.

## Синтаксис

### Операторы языка

**Основные** (statements) и **контекстные** (expressions)

**Основные** – операторы, с которых может начинаться новая строка.

**Контекстные** – операторы, которые могут использоваться внутри строки.

## Основные (statements)

### Условные операторы (Если>то>Иначе>Иначе если)

#### Синтаксис:

Если «контекстные операторы» то

*Новая строка с отступом и любой оператор из списка основных*

Иначе

*Новая строка с отступом и любой оператор из списка основных]*

#### Действие:

Условная конструкция, позволяющая совершить действия в зависимости от выполнения указанного условия. Допускает любое количество вложенных условий.

#### Пример1:

Если не существует ЕГРН.ДатаВыписки или она раньше 01.01.2001 то

Завершить Блок

Если необходимо явно указать действие в случае невыполнения условия, используется конструкция **Иначе**.

#### Пример2:

Если не существует ЕГРН.ДатаВыписки или она раньше 01.01.2001 то

Завершить Блок

Иначе

Сообщение "Все хорошо"

Также, если в случае невыполнения условия необходимо реализовать несколько различных сценариев, то для оптимизации можно использовать конструкцию **Иначе если**.

Допускается использование нескольких конструкций подряд.

**Важно:** если у вас много одноуровневых взаимоисключающих условий, рекомендуется использовать конструкции **Иначе если**, так как это ускорит работу кода.

#### Пример3:

Если ЕГРН.ДатаВыписки раньше 01.01.2001 то

Завершить Блок

Иначе если ЕГРН.ДатаВыписки равна 01.01.2001 то

Сообщение "хорошо"

Иначе если ЕГРН.ДатаВыписки позже 01.01.2001 то

Сообщение "Все хорошо"

### Записать

#### Синтаксис:

Записать «выражение\переменную» в «переменная»

#### Действие:

Записывает переданное значение в глобальную переменную, после чего новое значение доступно как в рамках текущего блока, так и во всех последующих. Если переменная находится внутри другой переменной, путь указывается полностью через точку.

Для корректной работы необходимо наличие глобальной переменной в глоссарии.

**Пример:**

Записать ЕГРН.Дата в НоваяПеременная

Записать "тестовая строка" в СписокПеременных.НоваяПеременная

Записать 20 в НоваяПеременная

Записать месяц от текущая дата в НоваяПеременная

## Запомнить

**Синтаксис:**

Запомнить «выражение\переменную» как «переменная» [типа «один из типов DSL»]

**Действие:**

Записывает переданное значение в локальную переменную, после чего новое значение доступно как в рамках текущего блока. Если переменная находится внутри другой переменной, путь указывается полностью через точку.

Опционально можно указать тип локальной переменной (по умолчанию - строка) для использования специфических свойств конкретного типа

**Пример:**

Запомнить ЕГРН.Дата как НоваяЛокальнаяПеременная

Запомнить "тестовая строка" как НоваяЛокальнаяПеременная.Текст

Запомнить 20 как НоваяЛокальнаяПеременная типа число

Записать "12.12.2012" как НоваяЛокальнаяПеременная типа дата

## Допуск сравнения

**Синтаксис:**

Допуск сравнения «число»

**Действие:**

Задаёт в блоке допустимое количество символов, с учетом которых будет проводиться сравнение в операторах "допустимо равно" и "выполняется сравнение имен". Указывается сразу после декларирования данных, перед первой проверкой.

**Пример:**

Используется ЕГРН

Допуск сравнения 1

Проверка "..."

## Обновить

**Синтаксис:**

Обновить «переменную»

**Действие:**

Сбрасывает значение переменной. **Важно:** переменная остается, к ней можно обратиться по имени, но значение будет отсутствовать. Для списочных переменных удаляются все элементы. Для несписочных вставляется значение None (нет значения). Может применяться как к локальным, так и к глобальным переменным.

**Пример:**

Записать 1 в НоваяПеременная (текущее значение 1)

Обновить НоваяПеременная (текущее значение - нет значения)

Создать список СписокУчастников

Записать "ФИО" в СписокУчастников (текущее значение - список с одним элементом)

Обновить СписокУчастников (текущее значение - пустой список)

## Вернуть

### Синтаксис:

Вернуть «выражение\переменную»

### Действие:

Оператор возвращает в линейный блок то значение/переменную, которая ему передана.

Используется только в функциональных блоках.

### Пример:

----- в линейном блоке

Запомнить результат выполнения ИмяФункциональногоБлока для ИмяАтрибута как НоваяПеременная

----- (В функциональном блоке)

Вернуть "Тест"

----- в линейном блоке

Сообщение НоваяПеременная ("Тест")

## Используется атрибут

### Синтаксис:

Используется атрибут «переменная»

### Действие:

Декларация данных в функциональном блоке. Оператор указывается первым в блоке. Имя атрибута должно совпадать с именем, которое использовалось при вызове из линейного блока, а сам атрибут маркируется префиксом *"Атрибут"*

После декларации атрибута, как и в случае с переменными в линейных блоках, можно обращаться ко всем его элементам.

### Пример:

----- в линейном блоке

Запомнить результат выполнения ИмяФункциональногоБлока для Ромашка как НоваяПеременная

----- (В функциональном блоке)

Используется атрибут Атрибут.Ромашка

## Создать список

### Синтаксис:

Создать список «название»

### Действие:

Создает локальную переменную, в которой можно хранить набор элементов, по умолчанию пустая

### Пример:

Создать список НоваяПеременная

## Добавить

### Синтаксис:

Добавить «выражение\переменную» в «переменная»

### Действие:

Добавляет элемент в список. Элемент всегда добавляется в конец списка. Добавить можно как значение, так и переменную

**Пример:**

Запомнить "Иванов Вася" как ФИО\_Подписанта

Создать список Тест

Добавить "Ура!" в Тест

Добавить 1 в Тест

Добавить 12.02.2020 в Тест

Добавить ФИО\_Подписанта в Тест

**Для каждого**

**Синтаксис:**

Для каждого «ПеременнаяЦикла» в\из «источник» выполнить

*Новая строка с отступом и любой оператор из списка основных*

**Действие:**

Позволяет поочередно перебрать каждый элемент в источнике и провести любые операции с его участием. Источником должна являться переменная типа "Список". Для возможности обращения к элементу, ему необходимо дать временное имя, поэтому в операторе используется переменная цикла - эта переменная живет только пока цикл не закончился.

Внутри цикла может инициировать новый цикл и так далее. Главное правило - нельзя в таких случаях давать одно и то же имя переменной циклов, так как это вызовет ошибку выполнения.

**Пример:**

Для каждого Участник в СписокУчастников выполнить

Если Участник равен "Иванов И.И." то

Сообщение "нашли"

Прервать

Иначе

Сообщение "Участник не подошел, идем дальше"

-----  
Для каждого Участник в СписокУчастников выполнить

Для каждого Лицо в СписокНовыхУчастников выполнить

Если Участник равен Лицо то

Сообщение "нашли совпадение участника в двух списках:" Участник

Прервать

**Вывод**

**Синтаксис:**

Вывод «выражение\переменная» «комментарий к выводу»

**Действие:**

Не создает каких-либо переменных, но формирует специальный элемент "Вывод", который учитывается при анализе результатов работы Блока правил и присутствует в ответе программы. Формулировки выводов строго согласованы со смежниками, комментарии могут быть любыми. Общий подход - наличие любого вывода в проверке означает, что проверка не завершилась штатно, и мы имеем какое отклонение по итогам проверки.

**Пример:**

Проверяем "Проверка ОПФ"

Если ЕГРЮЛ.ПНЮЛ содержит "ПАО" то

## Сообщение

### Синтаксис:

Сообщение «переменная\текст»

### Действие:

Информационный характер - добавляет в ответ программы информационные сообщения с указанным содержимым. Сообщения располагаются в рамках конкретной проверки ("Проверяем"). Необходимы для отладки программы. Данный оператор поддерживает математические операции, но не вносит никаких изменений в отображаемую информацию.

### Пример:

Запомнить "Запуск" как ИмяПеременной

Сообщение "Тест"

Сообщение ИмяПеременной

Сообщение 5

Сообщение ИмяПеременной + "Тест"

В последнем примере мы выведем в лог строку "ЗапускТест", однако переменную ИмяПеременной не изменим.

## Завершить блок

### Синтаксис:

Завершить блок

### Действие:

Останавливает выполнение блока, все глобальные данные передаются дальше в том виде, в котором они были на момент выполнения команды. Весь код, который находится ниже данной строки - выполняться не будет.

### Пример:

Завершить блок

## Завершить модуль

### Синтаксис:

Завершить модуль

### Действие:

Останавливает выполнение текущего блока, а также выполнение всех последующих линейных блоков в модуле. Если модуль является единственным в проекте, то в целом завершается весь цикл программы. Все глобальные данные передаются дальше в том виде, в котором они были на момент выполнения команды. Весь код, который находится ниже данной строки - выполняться не будет.

### Пример:

Завершить модуль

## Перейти в блок

### Синтаксис:

Перейти в блок «название\переменная»

### Действие:

Прерывает выполнение блока и вызывает новый блок. При этом возврата в текущий блок больше не происходит. Все глобальные данные передаются дальше в том виде, в котором они были на момент выполнения команды. Весь код, который находится ниже данной строки - выполняться не будет.



**Пример:**

Перейти в блок "Название блока"

----

Перейти в блок ПеременнаяИмяБлока

**Перейти в модуль****Синтаксис:**

Перейти в блок «название\переменная»

**Действие:**

Прерывает выполнение текущего блока и модуля. Вызывается указанный модуль. При этом возврата в текущий блок больше не происходит. Все глобальные данные передаются дальше в том виде, в котором они были на момент выполнения команды. Весь код, который находится ниже данной строки - выполняться не будет.

Отличие от *перейти в блок* в том, что прерывается выполнение всех блоков текущего модуля.

**Пример:**

Перейти в модуль "Название блока"

----

Перейти в модуль ПеременнаяИмяБлока

**Пройти дальше****Синтаксис:**

Пройти дальше

**Действие:**

Позволяет продолжить программу без каких-либо действий. Оператор в основном используется при отладке (Вы пишете условную или циклическую конструкцию, но еще не знаете, как будут выглядеть все ее части. Тогда их можно "заглушить", чтобы код можно было собрать и запустить).

**Пример:**

Для каждого Клиент в Клиенты

    Если не существует Клиент.Тип то

        Пройти дальше

    Иначе

        Сообщение Клиент

**Перейти к следующему****Синтаксис:**

Перейти к следующему

**Действие:**

При проведении циклических действий позволяет пропустить текущую итерацию и сразу перейти к следующей

**Пример:**

Для каждого Клиент в Клиенты

    Если не существует Клиент.Тип то

        Перейти к следующему

    Иначе

        Сообщение Клиент

**Прервать****Синтаксис:**

Прервать

**Действие:**

Прерывает выполнение циклической операции и передает управление в код, следующий после цикла.

**Пример:**

```
Для каждого Клиент в Клиенты
    Если не существует Клиент.Тип то
        Прервать
    Иначе
        Сообщение Клиент
```

**Проверяем****Синтаксис:**

Проверяем «название проверки»

**Действие:**

Создает новый раздел в блоке правил, который будет явно выделен в ответе. Фактически это "оглавление" блока правил. В качестве названия можно передать явный текст или название переменной. Количество проверок в блоке неограничено.

**Пример:**

```
Проверяем "Правоспособность подписантов"
....
....
Проверяем "Корпоративные ограничения"
```

**Исключить****Синтаксис:**

Исключить из "Имя переменной" совпадения с "Имя переменной"\значения

**Действие:**

Исключает из переменной типа Список все совпадения с указанными значениями. В качестве значений может выступить любой тип переменных, включая список.

**Пример:**

```
Исключить_из ОПФ совпадения_с СтопЛист
....
Исключить_из ОПФ совпадения_с "АО"
```

## Контекстные (expressions)

**выполняется проверка имен****Синтаксис:**

выполняется проверка имен «ФИО» и «ФИО»

**Действие:**

Выполняется проверка ФИО с учетом морфологии, падежей и инициалов. При этом в качестве каждого аргумента возможна передача как одной переменной, так и нескольких через запятую. ВАЖНО: учитывается уровень близости (см. основной оператор Допуск сравнения)

**Пример:**

```
Если выполняется проверка имен ИмяПеременной и ИмяВторойПеременной то...
....
Если выполняется проверка имен "Иванов Иван" и ИмяВторойПеременной то...
```

....

Если выполняется проверка имен ТД.ФИО\_Подписанта и ЕГРЮЛ.СвУчред.Имя, ЕГРЮЛ.СвУчред.Фамилия, ЕГРЮЛ.СвУчред.Отчество, то...

....

Если выполняется проверка имен "Шульга Сергей Александрович" и "Шульги", "Сергея", "Александровича", то...

### допустимо равны

#### Синтаксис:

допустимо равны «аргумент 1» и «аргумент 2»

#### Действие:

Выполняется сравнение двух аргументов с учетом допуска сравнения. В качестве аргументов можно передать переменную или строку. Оператор может работать только со строковым форматом.

ВАЖНО: учитывается уровень близости (см. основной оператор Допуск сравнения)

ВАЖНО: несмотря на похожую работу с оператором проверки имен, есть ряд ключевых отличий: допустимое сравнение не учитывает морфологию, падежи и инициалы, но позволяет сравнивать строки произвольной длины (в случае с ФИО количество слов в строке не может превышать 4-х).

#### Пример:

Если допустимо равны ИмяПеременной и ИмяВторойПеременной то...

....

Если допустимо равны "Иванов Иван" и ИмяВторойПеременной то...

....

Если допустимо равны "Шульга Сергей Александрович" и "Шульга С.А.", то...

### и

#### Синтаксис:

"условие" и "условие"..

#### Действие:

Два варианта:

- Логическое "и" связывает между собой два и более условий, используется в условных конструкциях
- используется для перечисления аргументов в отдельных операторах

#### Пример:

Если существует ЕГРЮЛ.ДатаРег и ЕГРЮЛ.ДатаВып раньше чем 12.12.2012 то...

....

Если допустимо равны "Иванов Иван" и ИмяВторойПеременной то...

### или

#### Синтаксис:

"условие" или "условие"..

#### Действие:

Логическое "или" связывает между собой два и более условий, используется в условных конструкциях. При этом, проверка условий идет последовательно слева направо, поэтому как только выполнится первое из условий в цепочке, связанной "или", проверка последующих "или" проводится уже не будет.

#### Пример:

Если существует ЕГРЮЛ.ДатаРег или ЕГРЮЛ.ДатаВып раньше чем 12.12.2012 то...

## равенство с аргументами (ОФП)

### Синтаксис:

"элемент1" равен с учетом ОПФ "элемент2"..

Оба элемента должны быть типа "строка"

### Действие:

Конструкция разработана для сравнения полных и сокращенных наименований юридических лиц, в том числе в составе предложения.

Два элемента проверяются на равенство с учетом организационно-правовой формы (ОПФ), при этом:

- учитываются принятые сокращения ОПФ
- не учитывается порядок ОПФ и самого названия - ООО "Ромашка" будет равно "Ромашка" ООО
- в случае, если сокращается часть наименования или целиком, такие случаи не могут быть проверены, анализ будет не валидный - ООО "Политическая Партия Свобода" будет не равно ООО "ПП Свобода"

Оператор возвращает логическое значение (Истина\Ложь)

### Пример:

Если ЕГРЮЛ.ПНЮЛ равно с учетом опф Устав.ЮЛ.Название то...

....

Если Устав.ЮЛ.Название не равно с учетом опф ЕГРЮЛ.ПНЮЛ то...

## результат выполнения

### Синтаксис:

Запомнить результат выполнения «название функционального блока» для «переменная» как «переменная2»

### Действие:

Вызывает выполнение Функционального блока с указанным названием, в который поступает на вход «переменная» и который возвращает результат выполнения в «переменная2».

### Пример:

Запомнить результат выполнения Расчет для Доход как Результат

## фильтрация (хотя бы для одного\для всех)

### Синтаксис:

..хотя бы для одного «переменная цикла» в «источник» «переменная цикла» «условие»..

..для всех «переменная цикла» в «источник» «переменная цикла» «условие»..

### Действие:

Конструкция проверяет «условие» для каждой «переменной цикла» в «источнике», и возвращает логическое значение (Истина\Ложь).

Для конструкции «хотя бы для одного» Истиной будет выполнение «условие» для одной и более переменных цикла

Для конструкции «для всех» Истиной будет выполнение «условие» для всех переменных цикла

### Пример:

Если хотя бы для одного Участник в Участники\_из\_Договоров\_оснований Участник.Источник содержит "ЕГРЮЛ" то

## результат поиска "переменная" в "выражении/переменная" в котором "выражение"

### Синтаксис:

.. результат поиска «переменная цикла» в «источник» в котором «условие»..

**Действие:**

Конструкция проверяет «условие» для каждой «переменной цикла» в «источнике» и запоминает те «переменные цикла» для которых «условие» истинно.

**Пример:**

Запомнить результат поиска Участник в Участники в котором существует Участник.Тип и он равен "Физическое лицо" как Продавцы\_Физ

**период от "переменная/значение" до "переменная/значение"****Синтаксис:**

.. период «дата1» до «дата2» «условие»..

**Действие:**

Конструкция проводит расчет разницы между «дата2» и «дата1» по правилам для типа данных Период (см. подробнее в описании типа данных Период).

**Пример:**

Если период от Участник.ДатаРождения до Сделка.ДатаРегистрации меньше 18 лет то

**не****Синтаксис:**

.. не "условие"..

**Действие:**

Логическое "не" меняет критерий истинности или ложности следующего условия.

**Пример:**

Если хотя бы для одного Участник в Участники\_из\_Договоров\_оснований Участник.Источник не содержит "ЕГРЮЛ" то

Запомнить результат поиска Участник в Участники в котором не существует Участник.Тип и он равен "Физическое лицо" как Продавцы\_Физ

**Равенство****Синтаксис:**

.. «переменная1» равен "переменная2"..

**Действие:**

Производится сравнение двух значений которые могут быть заданы как переменной, так и значениями напрямую. Для различных типов учитывает специфику данных. Например, для ФИО не будет учитываться регистр.

Оператор возвращает логическое значение (Истина\Ложь)

**Пример:**

Если Участник.ФИО равен Покупатель.ФИО то

**меньше или равен****Синтаксис:**

.. «переменная1» меньше или равен "переменная2"..

**Действие:**

Производится математическая операция сравнения двух значений типа Дата/Число/Деньги/Период.

Оператор возвращает логическое значение (Истина\Ложь)

**Пример:**

Если Покупатель.Возраст меньше или равен 18 то

### **больше или равен**

**Синтаксис:**

.. «переменная1» больше или равен "переменная2"..

**Действие:**

Производится математическая операция сравнения двух значений типа Дата/Число/Деньги/Период.

Оператор возвращает логическое значение (Истина\Ложь)

**Пример:**

Если ЕГРЮЛ.КоличествоСобственникиДома больше или равен 2 то

### **раньше или равен**

**Синтаксис:**

.. «переменная1» раньше или равен "переменная2"..

**Действие:**

Производится математическая операция сравнения двух значений типа Дата.

Оператор возвращает логическое значение (Истина\Ложь)

**Пример:**

Если ЕГРЮЛ.ДатаПриватизации раньше или равен 01.01.2002 то

### **позже или равен**

**Синтаксис:**

.. «переменная1» позже или равен "переменная2"..

**Действие:**

Производится математическая операция сравнения двух значений типа Дата.

Оператор возвращает логическое значение (Истина\Ложь)

**Пример:**

Если ДомоваяКнига.Дата позже или равен текущая дата то

### **раньше чем..**

**Синтаксис:**

.. «переменная1» раньше чем "переменная2"..

**Действие:**

Производится математическая операция сравнения двух значений типа Дата.

Оператор возвращает логическое значение (Истина\Ложь)

**Пример:**

Если ДомоваяКнига.Дата раньше, чем текущая дата то

### **позже чем..**

**Синтаксис:**

.. «переменная1» позже чем "переменная2"..

**Действие:**

Производится математическая операция сравнения двух значений типа Дата.

Оператор возвращает логическое значение (Истина\Ложь)

**Пример:**

Если ДомоваяКнига.Дата позже, чем текущая дата то

### **содержит**

**Синтаксис:**

.. «переменная1» содержит "значение"..

**Действие:**

Производится поиск значения или переменной в «переменная1». Условие выполняется если есть «переменная1» содержит искомое значение. При поиске учитываются склонения и множественность.

Оператор возвращает логическое значение (Истина\Ложь)

**Пример:**

Если ЕГРЮЛ.Адрес содержит «Россия» то

**содержит посимвольно****Синтаксис:**

.. «переменная1» содержит "значение"..

**Действие:**

Производится поиск значения или переменной в «переменная1». Условие выполняется если есть переменная1 содержит искомое значение. Поиск производится посимвольно, символы значения должны содержаться в переменной ровно в той же последовательности и виде, как указаны в искомом значении.

Оператор возвращает логическое значение (Истина\Ложь)

**Пример:**

Если ЕГРЮЛ.Адрес содержит «Рос» то

**строго равно****Синтаксис:**

.. «переменная1» строго равна «переменная2»..

**Действие:**

Для переменных типа Строка позволяет выполнить посимвольное сравнение.

**Пример:**

Если Выписка.Покупатель.ФИО строго равно Договор.Покупатель.ФИО то

**может использоваться как..****Синтаксис:**

Если «ИмяПеременной» может использоваться как Период..

**Действие:**

Проверяет возможность использовать(представить) переменную в нужном типе. Возвращает логическое значение. В случае истинности к переменной можно применить методы нужного типа. Например, можно строку «2 недели» обрабатывать как переменную типа Период.

**Пример:**

Запомнить «12 декабря 2021 года» как Срок

Если Срок может использоваться как Дата то

Запомнить Срок + 5 дней как НовыйСрок

+

**Синтаксис:**

.. «переменная1» + «переменная2»..

**Действие:**

Для чисел и строк работает по-разному: для чисел это математическая операция сложения, для строк – конкатенация.

**Пример:**

Запомнить Стоимость + 5 как Сумма

Запомнить Покупатель.Фамилия + Покупатель.Имя + Покупатель.Отчество как ФИО

-

**Синтаксис:**

.. «переменная1» - «переменная2»..

**Действие:**

Производит математическую операцию вычитания.

**Пример:**

Запомнить Стоимость - 5 как Разница

\*

**Синтаксис:**

.. «переменная1» \* «переменная2»..

**Действие:**

Производит математическую операцию умножения.

**Пример:**

Запомнить Зарплата \* 0,87 как Доход

/

**Синтаксис:**

.. «переменная1» / «переменная2»..

**Действие:**

Производит математическую операцию деления.

**Пример:**

Запомнить Стоимость / 2 как Оплата

**скобочные выражения****Синтаксис:**

.. «переменная1» / («переменная2» + «переменная3»)..

**Действие:**

Определяют приоритет выполнения математических операций – операции в скобках имеют приоритет над операциями вне скобок.

**Пример:**

Запомнить  $(100 - 13) * (Зарплата / 100)$  как Доход

**любой из..****Синтаксис:**

.. «переменная1» «условие» любому из «значение1», «значение2»..

**Действие:**

Производит проверка выполнения условия для переменной1 и любого из указанных значений, переменных, списка.

**Пример:**

Если Участник.Источник равен любому из «договор купли-продажи», «мена», «наследство» то

**ни один из..**



**Синтаксис:**

.. «переменная1» «условие» ни один из «значение1», «значение2»..

**Действие:**

Производит проверку выполнения условия для переменной1 и любого из указанных значений, переменных, списка. Возвращает «истина», если условие не выполняется ни для одного значения.

**Пример:**

Если НазначениеОбъектаНедвижимости не содержит посимвольно ни один из "газопровод", "водопровод", "трубопровод" то

**элемент даты (день/месяц/год от даты)****Синтаксис:**

день\месяц\год от «ИмяПеременной»

**Действие:**

Возвращает запрошенный элемент из Переменной типа Дата. Возвращаемое значение является Переменной типа Число.

**Пример:**

Если месяц от Договор.ДатаПодписания равен 6 то...

**округление****Синтаксис:**

округление к большему «ИмяПеременной»

**Действие:**

Округляет Переменные типа «Число» и «Деньги» к большему или меньшему целому

**Пример:**

Если округление к меньшему Договор.СуммаСделки больше 1000 то...

**анализ ФИО (имя/фамилия/отчество в ФИО)****Синтаксис:**

имя в фео\фамилия в фео\отчество в фео «ИмяПеременной»..

**Действие:**

Выделяет в строке, содержащей ФИО, требуемый элемент и возвращает его. В случае если нужного элемента не найдено, возвращается None (пустое значение). Это бывает, когда мы ищем отчество в строке вида «Иванов Сергей».

**Пример:**

Если фамилия в фео ДанныеПокупателя равна ЕГРЮЛ.ФЛ.Фамилия то...

**местоимения****Синтаксис:**

«его\ее\он\она\они»

**Действие:**

Позволяет в рамках условной конструкции «Если» после первого указания Переменной, далее по блоку использовать вместо имени местоимение, что позволяет сократить код и сделать его более читаемым.

**Пример:**

Если существует ЕГРЮЛ.ДатаРегистрации и она раньше «12.01.01» то  
Запомнить ее как ПерваяДата

**сумма**

**Синтаксис:**

сумма "ИмяПеременной"

**Действие:**

Возвращает сумму элементов списка, которые поддерживают операции больше/меньше (числа, деньги, периоды)

**Пример:**

Если сумма ВзносыУставКап больше 10000 то...

**максимальный****Синтаксис:**

максимальный в/из "ИмяПеременной"

**Действие:**

Возвращает самое большое число из переменной типа Список, в котором лежат только числовые элементы и элементы, к которым применимы операции больше/меньше. Например, деньги или периоды.

**Пример:**

Запомнить максимальный в СписокЧисел как СамоеБольшоеЧисло

**минимальный****Синтаксис:**

минимальный в/из "ИмяПеременной"

**Действие:**

Возвращает самое маленькое число из переменной типа Список, в котором лежат только числовые элементы и элементы, к которым применимы операции больше/меньше. Например, деньги или периоды.

**Пример:**

Запомнить минимальный в СписокЧисел как СамоеМаленькоеЧисло

**самый поздний****Синтаксис:**

самый поздний в/из "ИмяПеременной"

**Действие:**

Возвращает самую позднюю дату из переменной типа Список, в котором лежат только элементы даты.

**Пример:**

Запомнить самый поздний в СписокДат как ПоследнееОбновление

**самый ранний****Синтаксис:**

самый ранний в/из "ИмяПеременной"

**Действие:**

Возвращает самую раннюю дату из переменной типа Список, в котором лежат только элементы даты.

**Пример:**

Запомнить самый ранний в СписокДат как ПерваяЗапись

**первый****Синтаксис:**

первый в/из "ИмяПеременной"

**Действие:**

Возвращает первый по индексу элемент списка (или по-другому - элемент, который был добавлен в список первым по времени). Работает только со списками. Если список пуст, значение не возвращается.

**Пример:**

Запомнить первый в СписокУчастников как КрайнийУчастник

**последний**

**Синтаксис:**

последний в/из "ИмяПеременной"

**Действие:**

Возвращает последний по индексу элемент списка (или по-другому - элемент, который был добавлен в список последним по времени). Работает только со списками. Если список пуст, значение не возвращается.

**Пример:**

Запомнить последний из СписокУчастников как КрайнийУчастник

**пустой**

**Синтаксис:**

пустой «ИмяПеременной»

**Действие:**

Проверяет наличие элементов в списке, возвращает Истину (True), если элементов больше 0

**Пример:**

Если пустой СписокУчастников то...

**существует**

**Синтаксис:**

существует "ИмяПеременной"\ "выражение"

**Действие:**

Проверяет существование локальной или глобальной переменной. Для целей DSL существующей является переменная с действительным значением, то есть переменная, которой присвоено какое-либо значение, и оно отличается от None. К примеру, у нас есть глобальная переменная СоставУчастников, но к моменту проверки в нее еще ни разу ничего не записывали - она не существует.

В качестве переменной оператор может принимать выражение, к которому будет применена проверка

**Пример:**

Если существует СоставУчастников то...

....

Если существует любой из Проблема\_в\_подписи\_ОИГ, НеудКач\_ОИГ то...

**количество**

**Синтаксис:**

количество «ИмяПеременной»

**Действие:**

Возвращает количество элементов в указанной переменной. Работает только со списками.

**Пример:**

Если количество СписокУчастников больше 4 то...

#### четное

**Синтаксис:**

четное "ИмяПеременной"

**Действие:**

Возвращает Истину (true), если значение переменной является четным. Может использоваться для чисел, периодов и денег.

**Пример:**

Если четное ЧислоУчастников то...

#### текущая дата

**Синтаксис:**

текущая дата

**Действие:**

Возвращает текущую дату

**Пример:**

Если ДатаСоздания больше текущая дата то...

## Создание работающей программы на примере

Для примера создадим программу по упрощенному расчету НДФЛ

Необходимо определиться с входными данными, заложенными в глоссарий  
В нашем случае это будут:

Данные\_по\_доходам - тип Словарь

Доходы - тип Список (так как может быть много различных доходов)

Доход - тип Словарь

Сумма\_дохода - тип Строка

Валюта\_дохода - тип Строка с заранее заданным списком значений

Тип\_дохода - тип Строка с заранее ограниченным списком значений

Курс\_Валют - тип Список (содержит все ожидаемые валюты)

Валюта - тип Словарь (содержит данные по конкретной валюте)

Название - тип Строка

Курс - тип Строка

На выходе мы отдадим Переменная.Заключение в котором будет содержаться сумма выплаты по НДФЛ

Сама программа будет содержать основной блок (для подсчета итогового НДФЛ) и функциональный блок (для расчета НДФЛ по каждому типу доходов)

#### Основной блок

Блок "Расчет НДФЛ" !Декларируем название основного блока правил

Используются Данные\_по\_доходам

Используются Переменная

!Декларируем используемые глобальные переменные

**Проверяем "Расчет НДСЛ"** !Декларируем операцию которая осуществляется частью блока правил (в каждом блоке может быть множество операций

Если не пустой Данные\_по\_доходам.Доходы то

Запомнить 0 как СуммаНДСЛ

Для каждого Доход в Данные\_по\_доходам.Доходы

Запомнить результат выполнения Расчет для Доход как Результат !Четырьмя строками выше мы проверили наличие данных по доходам (на случай если они не пришли), создали Локальную переменную в которой будем суммировать НДСЛ и запустили цикл который для каждого отдельного дохода вызовет Функциональный блок "Расчет". По итогам работы функционального блока будет возвращена Локальная переменная "Результат" в которой будет сумма расчета

Если Доход.Валюта\_дохода не равна "рубль" то

Запомнить результат поиска Нужная\_валюта в Данные\_по\_доходам.Курсы\_Валют в котором существует Нужная\_валюта.Валюта.Название и он равен Доход.Валюта\_дохода как Искомая\_валюта

Запомнить СуммаНДСЛ + Результат \* Искомая\_валюта.Курс как СуммаНДСЛ

Иначе

Запомнить СуммаНДСЛ + Результат как СуммаНДСЛ

!Здесь мы совершили ряд операций - проверили валюту в которой был указан доход. Если эта валюта не является рублем, то мы запустили поиск соответствующей валюты во входных данных, нашли её и умножили её курс на полученную из Функционального блока сумму НДСЛ, после чего добавили её к локальной переменной в которой считаем итоговую сумму. Если же валюта была изначально в рублях - просто добавили расчет к итоговой сумме.

Записать СуммаНДСЛ в Переменная.Заключение

**Завершить блок**

!После выполнения цикла расчета для всех доходов - передаем итоговую сумму НДСЛ далее и завершаем программу

Иначе

Записать "Отсутствуют данные по доходам" в Переменная.Заключение

**Завершить блок**

!Сообщаем о том что данные о доходах в программу не поступили

**Функциональный блок**

Блок "Расчет" как Расчет

Используются Доход

Используется параметр Доход

!Декларируем используемые входные данные в блок и название блока

Если Доход.Тип\_дохода равен любому из “получения дивидендов до 2015 года”, “получения процентов по облигациям с ипотечным покрытием, эмитированным до 1 января 2007 г.”, “получения доходов учредителями доверительного управления ипотечным покрытием. Такие доходы должны быть получены на основании приобретения ипотечных сертификатов участия, выданных управляющим ипотечным покрытием до 1 января 2007 г” то

Запомнить 9 как ПроцентнаяСтавка

Если Доход.Тип\_дохода равен “дивиденды, полученные от российских организаций физическими лицами, которые не являются налоговыми резидентами РФ.” то

Запомнить 15 как ПроцентнаяСтавка

Если Доход.Тип\_дохода равен любому из “выигрышей и призов, получаемых в проводимых конкурсах, играх и других мероприятиях в целях рекламы товаров, работ и услуг, в части превышения установленных размеров”, “процентных доходов по вкладам в банках в части превышения установленных размеров”, “суммы экономии на процентах при получении налогоплательщиками заемных (кредитных) средств в части превышения установленных размеров”, “платы за использование денежных средств членов кредитного потребительского кооператива (пайщиков), а также процентов за использование сельскохозяйственным кредитным потребительским кооперативом средств, привлекаемых в форме займов от членов сельскохозяйственного кредитного потребительского кооператива или ассоциированных членов сельскохозяйственного кредитного потребительского кооператива, в части превышения установленных размеров” то

Запомнить 35 как ПроцентнаяСтавка

Если не существует ПроцентнаяСтавка то

Запомнить 15 как ПроцентнаяСтавка

!В упрощенном виде и без учета максимальных размеров и резидентства мы определяем по тексту Тип\_дохода подходящую процентную ставку и создаем локальную переменную которая её содержит

Запомнить  $\text{Доход.Сумма\_дохода} / 100 * \text{ПроцентнаяСтавка}$  как Результат

Вернуть Результат

!Проводим математические операции по расчету НДФЛ, запоминаем их как переменную и передаем обратно в Основной блок правил