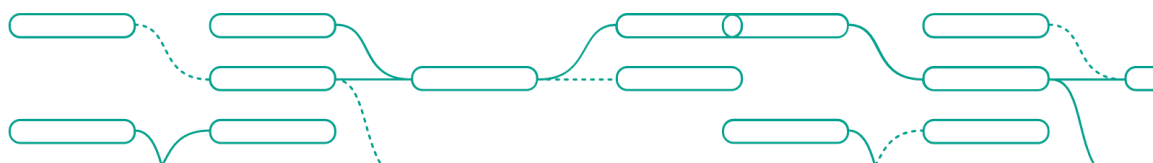


Описание эксплуатации программного обеспечения
Arena Data Catalog Data Quality Framework
(ADC.DQF)

Москва
2025



Оглавление

1	Сценарии взаимодействия с приложением	3
1.1	Владелец процесса качества данных	3
1.2	Аналитик качества данных	3
1.3	Владелец источника данных	4
1.4	Потребитель данных	4
2	Редактор правил	4
2.1	Настройка правил	5
2.1.1	Базовый алгоритм	5
2.1.2	Механизм управления потоком	8
2.1.3	Правило	9
2.2	Проверка корректности правила	10
2.3	Меню редактора правил	10
3	Каталог правил	12
3.1	Список правил	12
3.2	Список групп	12
3.3	Создание группы	13
4	Планирование задач	15
4.1	Список задач	15
4.2	Создание задачи	16
4.3	Изменение статуса задачи	16
4.4	Просмотр результатов выполнения задачи	17
5	Отчетность	19
5.1	Формирование и загрузка отчетов	20
6	Дашборды	21
6.1	Добавление и изменение виджетов	22
6.2	Редактирование дашборда	24

1 Сценарии взаимодействия с приложением

Платформа предоставляет широкий набор возможностей для работы с качеством данных, охватывая все этапы — от проектирования проверок до анализа их результатов. В зависимости от роли пользователя в организации, сценарии взаимодействия с приложением могут существенно различаться.

В этом разделе представлены типовые кейсы использования системы с учётом ролей участников процесса. Они помогут лучше понять, как отдельные компоненты платформы могут использоваться для решения практических задач — от обеспечения корректности данных до управления стратегией качества на уровне всей компании.

1.1 Владелец процесса качества данных

Задачи: отвечает за формирование и контроль выполнения политики качества данных в организации. Участвует в согласовании правил и метрик, принимает управленческие решения на основе данных.

Сценарий 1: «Мониторинг реализации политики качества данных».

1. Заходит в дашборд, чтобы оценить динамику числа ошибок по ключевым группам проверок за месяц;
2. Делает drill-down до конкретных правил, где наблюдается рост количества несоответствий;
3. Формирует сводный отчет по этим правилам за последние 3 месяца для анализа в Excel;
4. На основе данных и трендов инициирует изменение набора правил или уточнение логики.

1.2 Аналитик качества данных

Задачи: проектирует и реализует правила проверки данных, отвечает за их корректность и актуальность, участвует в анализе результатов проверок и их интерпретации.

Сценарий 1: «Разработка и тестирование нового правила проверки».

1. Создает новое правило в редакторе, используя базовые алгоритмы и механизмы управления потоком;
2. Добавляет вложенное правило и параметризует все блоки правила;
3. Вводит пример входных данных и запускает проверку правила;
4. Проверяет выходной результат, выявляет ошибку в логике, исправляет выражение в одном из базовых алгоритмов;
5. Сохраняет правило в каталог, добавляет его в соответствующую группу.

Сценарий 2: «Актуализация существующих проверок».

1. Открывает правило из каталога, проверяет его соответствие обновленной бизнес-логике;
2. Вносит изменения в правило;
3. Сохраняет новую версию правила;
4. Использует тестирование правила, чтобы убедиться в корректности результата.

1.3 Владелец источника данных

Задачи: отвечает за предоставление и корректность данных, передаваемых в систему. Его цель — своевременно выявлять и устранять несоответствия на своей стороне.

Сценарий 1: «Контроль за качеством перед выгрузкой данных».

1. Создает задачу проверки данных, загружая массив объектов из своей системы;
2. Выбирает нужную группу правил;
3. Отслеживает статус задачи, затем просматривает детализированные результаты выполнения;
4. Находит объекты, не прошедшие проверку, выгружает json с доп. информацией об ошибках;
5. Вносит изменения в источник данных перед публикацией информации.

1.4 Потребитель данных

Задачи: использует данные в своей работе (например, для отчетности, аналитики или операционных решений). Заинтересован в получении проверенной и корректной информации.

Сценарий 1: «Проверка данных перед использованием в отчете».

1. Пользователь открывает ранее созданный дашборд, содержащий информацию о качестве данных;
2. Добавляет на панель виджет в формате столбчатой диаграммы, отображающий количество непройденных проверок с разбиением по группам правил;
3. Анализирует результаты и убеждается, что по интересующим его группам данных за последний период не зафиксировано критичных нарушений;
4. Использует проверенные и надежные данные для подготовки внешнего BI-отчета или включения в аналитическую презентацию.

2 Редактор правил

Редактор правил — это визуальный инструмент для создания, просмотра и редактирования правил, описывающий логику обработки и анализа данных (Рисунок 1). Он представляет графический интерфейс, позволяющий пользователям без навыков программирования легко формировать логические условия, определяющие, каким требованиям должны соответствовать данные.



Рисунок 1 - Редактор правил

2.1 Настройка правил

Правило – структурированная последовательность шагов (базовых алгоритмов, механизмов управления потоком и других правил), где каждый шаг возвращает статус выполнения и оперирует данными через контекст исполнения для решения конкретной бизнес-задачи.

Пользователь может формировать структуру правила, добавляя или удаляя отдельные блоки. Каждый блок представляет собой либо базовый алгоритм, либо механизм управления потоком, либо вызывает выполнение другого сконфигурированного правила.

2.1.1 Базовый алгоритм

Базовый алгоритм - минимальная логическая конструкция, определяющая основную операцию проверки, получения или трансформации данных.

Для понимания возможностей проверки данных ниже (Таблица 1) приведена таблица с перечнем доступных базовых алгоритмов, которые могут быть включены в правило.

Название	Идентификатор	Описание
Проверить значение	CheckValue	Проверка и сравнение значений атрибутов
Присвоить значение	SetValue	Установка значения атрибута
Преобразовать массив	Collect	Сбор и фильтрация данных
Добавить в массив	AppendValue	Добавление атрибута в список
Продолжить	Continue	Продолжить исполнение
Содержится в справочнике	CheckValueFromDictionary	Проверка значения на вхождение в справочник
GraphQL запрос	GraphQLQuery	Запрос связанных объектов из GraphQL-сервиса
HTTP запрос	HttpRequest	Вызов HTTP методов

JDBC запрос	JDBCRequest	Выполнение SQL-запроса к поддерживаемым СУБД через стандартный интерфейс JDBC.
Проверить СНИЛС	CheckSnils	Проверка СНИЛС подсчетом контрольного числа в соответствии с алгоритмом, присвоения маркера качества
Проверить ИНН физического или юридического лица	CheckInn	Проверка идентификационного номера налогоплательщика (далее – ИНН) подсчетом контрольного числа в соответствии с алгоритмом для физических или юридических лиц (зависит от конфигурации базового алгоритма), присвоение маркера качества
Проверить ОГРН	CheckOgrn	Проверка и очистка государственного регистрационного номера записи о создании юридического лица либо записи о первом представлении сведений о юридическом лице (ОГРН)
Проверить адрес ¹	CheckAddress	Проверка адреса на вхождение в Государственный адресный реестр (далее – ГАР)

Таблица 1 - Список базовых алгоритмов

Ниже представлен пример вызова базового алгоритма "Добавить в массив" (AppendValue) в yaml-формате, который помещает элемент из пути `$.variables.recordNumber` в массив по пути `$.variables.recordNumbersWithErrors`.

```
AppendValue:
  expr: $.variables.recordNumber
  result: $.variables.recordNumbersWithErrors
```

YAML-описание выше соответствует следующему визуальному элементу в редакторе (Рисунок 2).

¹ Требуется подключение продукта «Гражданский Фактор» ООО «Клин Дейта» или «Фактор» ООО «ХФ Лабс».

Добавить в массив

dsl

numeric

literal

array

object

Элемент:

\$.variables.recordNumber

dsl

numeric

literal

array

object

Массив:

\$.variables.recordNumbersWithErrors

Рисунок 2 - Базовый алгоритм "Добавить в массив"

2.1.1.1 Типы полей

В визуальном представлении правила доступен выбор типа поля из набора predefined значений (Таблица 2).

Тип поля	Наименование	Описание
dsl	Выражение	В выражении могут использоваться: - Константы любых типов, поддерживаемых в формате JSON — строки, числа, логические значения, null, объекты и массивы; - JSONPath-селекторы, позволяющие обращаться как к данным, переданным на вход, так и к данным, заданным внутри алгоритма; - Функции и операторы, с помощью которых можно проводить вычисления, задавать условия или обрабатывать данные. Такие конструкции могут быть вложенными и использовать другие выражения.
array	Массив	Упорядоченный список значений.
object	Объект	Набор пар «ключ: значение».
numeric	Число	Целое или дробное значение.
literal	Литерал	Специальные значения: true, false, null.

Таблица 2 - Типы полей

Отдельного внимания заслуживает тип «выражение» (dsl), используемый для вычисляемых условий. В выражениях можно использовать множество операторов и функций, поддерживаемых DQF.

Полный список доступных операторов и функций представлен в сопроводительной документации.

2.1.2 Механизм управления потоком

Механизм управления потоком — это связующее звено, объединяющее базовые алгоритмы в единую последовательность исполнения – правило.

Ниже перечислены доступные механизмы управления потоком, которые могут использоваться при настройке правила.

Название	Идентификатор	Описание
Выполнить все	all-of	Объединение последовательного исполнения шагов.
Повторять для	for-each	Итерирование по коллекции.
Повторять пока	while	Цикл, пока выполняется условие.
Иначе	otherwise	Альтернативная ветвь исполнения. <i>Не добавляется как отдельный самостоятельный блок, а прикрепляется к существующему элементу в редакторе, у которого предусмотрена логика разветвления.</i>

Таблица 3 - Механизмы управления потоком

Ниже (Блок кода 1) представлен пример вызова механизма управления потоком «Повторять пока» (while), внутри которого происходит вызов базового алгоритма «Присвоить значение» (SetValue).

```
do-at-least-once:
  SetValue:
    expr: $.variables.x - $.variables.n
    result: $.variables.x
  while: $.variables.x > 0
```

YAML-описание выше соответствует следующему визуальному элементу в редакторе (Рисунок 3).

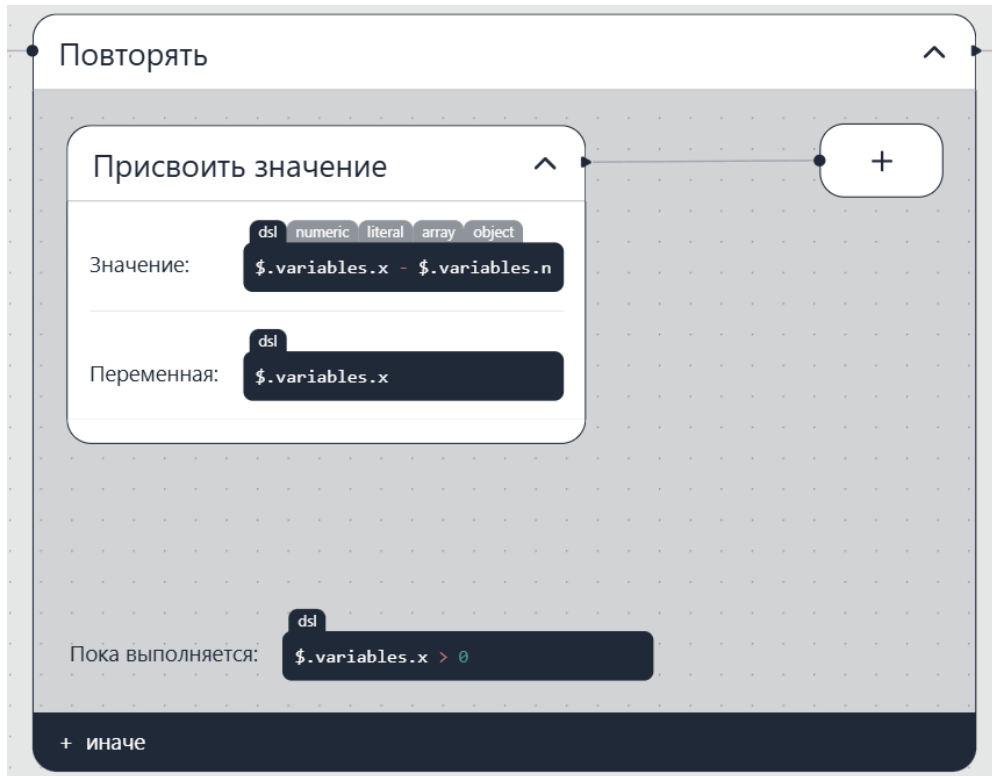


Рисунок 3 - Вызов механизма управления потоком «Повторять пока» с базовым алгоритмом "Присвоить значение"

2.1.3 Правило

При конфигурации правила, помимо выбора базовых алгоритмов и механизмов управления потоком, можно также использовать вызов другого правила. В этом случае во визуальном редакторе правило выбирается из каталога, а в текстовом формате необходимо указать его идентификатор вручную.

Ниже приведен пример вызова правила с идентификатором `bank_metrics`: сначала в формате YAML, затем его визуальное представление в редакторе (Рисунок 4).

```
- sample_algo: {}
```

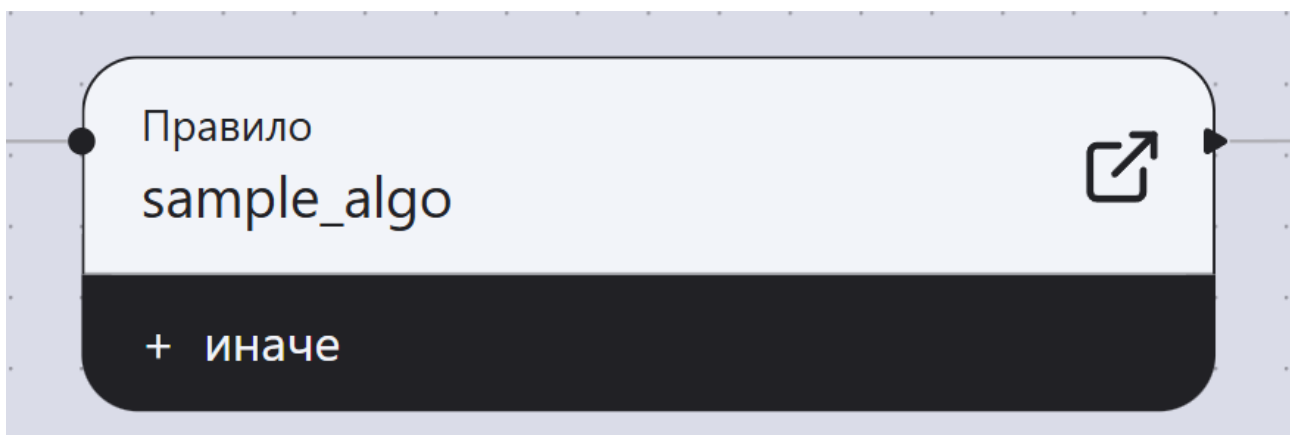


Рисунок 4 - Вызов правила sample_algo

2.2 Проверка корректности правила

Редактор правил предоставляет возможность протестировать правило непосредственно в интерфейсе (Рисунок 5). Пользователь может вручную задать входные данные и выполнить проверку, чтобы увидеть результат выполнения правила — включая итоговый статус и возвращаемые данные (Рисунок 6). Это особенно полезно на этапе разработки и отладки правил, позволяя оперативно выявить ошибки конфигурации или логики.

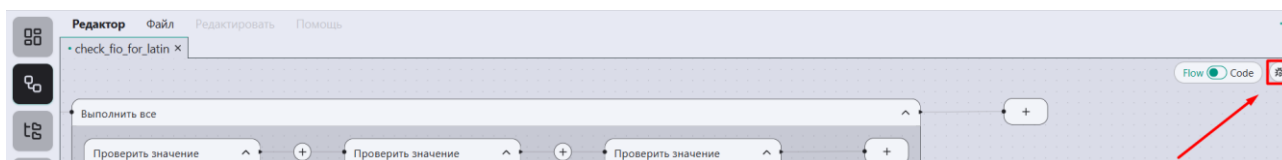


Рисунок 5 – Переход к проверке правила на тестовых данных

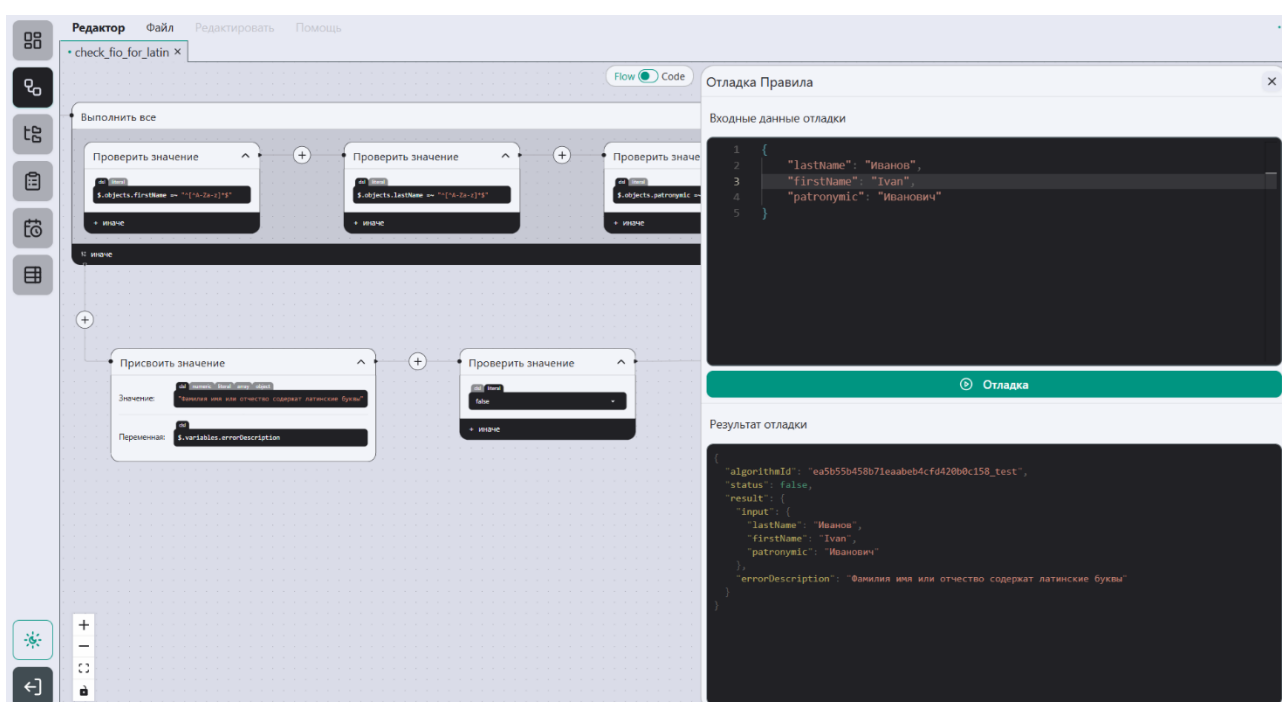


Рисунок 6 - Результат выполнения тестового запуска

2.3 Меню редактора правил

Меню редактора предоставляет доступ к основным действиям, связанным с управлением правилами (Рисунок 7). Через него пользователь может выполнять следующие операции:

- Открыть правило — выбрать любое существующее правило из каталога и загрузить его в редактор для просмотра или изменения;
- Создать новое правило — начать конфигурацию нового правила в новой вкладке редактора;
- Сохранить — сохранить изменения, внесённые в текущее правило;
- Сохранить как — сохранить правило под другим идентификатором, создав его копию;
- Экспортировать — сохранить текущее правило в виде YAML-файла;
- Удалить — удалить открытое в редакторе правило.

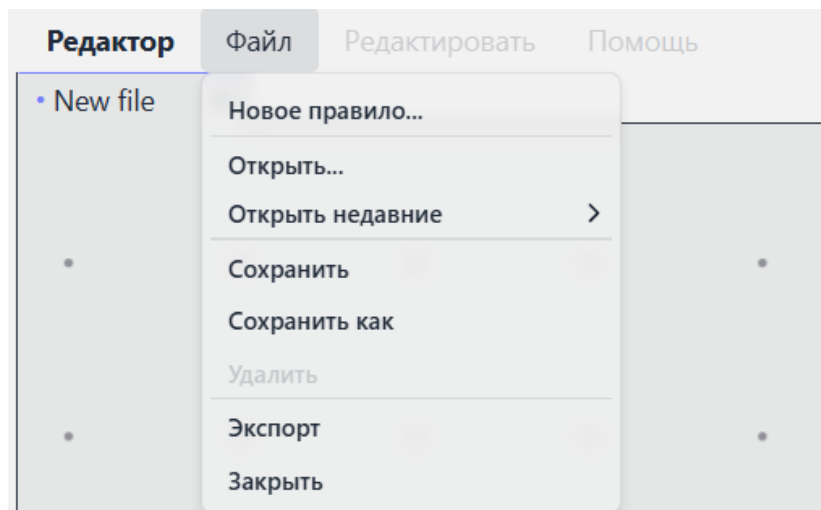


Рисунок 7 - Меню редактора правил

Все действия выполняются через графический интерфейс и доступны в верхней панели приложения.

3 Каталог правил

Каталога правил – это сервис, позволяющий пользователю управлять правилами и группами правил.

3.1 Список правил

В каталоге правил список представляется в виде таблицы с ключевыми параметрами каждого правила (Рисунок 8). Пользователи могут управлять этим списком, фильтруя и сортируя его по необходимым параметрам. Кроме того, с этой страницы можно открыть правило в редакторе, удалить выбранное правило или создать новое, с автоматическим переходом в редактор правил.



название правила	версия	создано	дата создания	обновлено	дата обновления
bank_metrics	9	dqf-cas-updater	2025-07-01 10:33	dqf-cas-updater	2025-07-08 10:40
check_fio_for_latin	1	test-user	2025-07-02 11:24	test-user	2025-07-02 11:24
check_fio_for_len	1	test-user	2025-07-02 11:25	test-user	2025-07-02 11:25
checking_person_details	9	dqf-cas-updater	2025-07-01 10:33	dqf-cas-updater	2025-07-08 10:40
check_living_area_cross	2	test-user	2025-07-02 11:26	test-user	2025-07-02 11:27
check-param_value	1	test-user	2025-07-10 18:39	test-user	2025-07-10 18:39
get_closest_fri	9	dqf-cas-updater	2025-07-01 10:33	dqf-cas-updater	2025-07-08 10:40
jdbc-test	4	test-user	2025-07-03 16:46	test-user	2025-07-09 18:12
marketing_metrics_analysis	9	dqf-cas-updater	2025-07-01 10:33	dqf-cas-updater	2025-07-08 10:40
random-rule	9	dqf-cas-updater	2025-07-01 10:33	dqf-cas-updater	2025-07-08 10:40
sample_algo	3	test-user	2025-07-09 16:34	test-user	2025-07-09 17:24
sample-algo-date-date	9	dqf-cas-updater	2025-07-01 10:33	dqf-cas-updater	2025-07-08 10:40
sample-algo-date-timestamp	9	dqf-cas-updater	2025-07-01 10:33	dqf-cas-updater	2025-07-08 10:40
sample-algo-false	9	dqf-cas-updater	2025-07-01 10:33	dqf-cas-updater	2025-07-08 10:40
sample-algo-fraction-fraction	9	dqf-cas-updater	2025-07-01 10:33	dqf-cas-updater	2025-07-08 10:40
sample-algo-jdbc	2	test-user	2025-07-01 10:48	test-user	2025-07-01 11:23
sample-algo-number-date	9	dqf-cas-updater	2025-07-01 10:33	dqf-cas-updater	2025-07-08 10:40
sample-algo-number-decimal	9	dqf-cas-updater	2025-07-01 10:33	dqf-cas-updater	2025-07-08 10:40
sample-algo-number-int	9	dqf-cas-updater	2025-07-01 10:33	dqf-cas-updater	2025-07-08 10:40
sample-algo-number-string	9	dqf-cas-updater	2025-07-01 10:33	dqf-cas-updater	2025-07-08 10:40

Рисунок 8 - Каталог правил

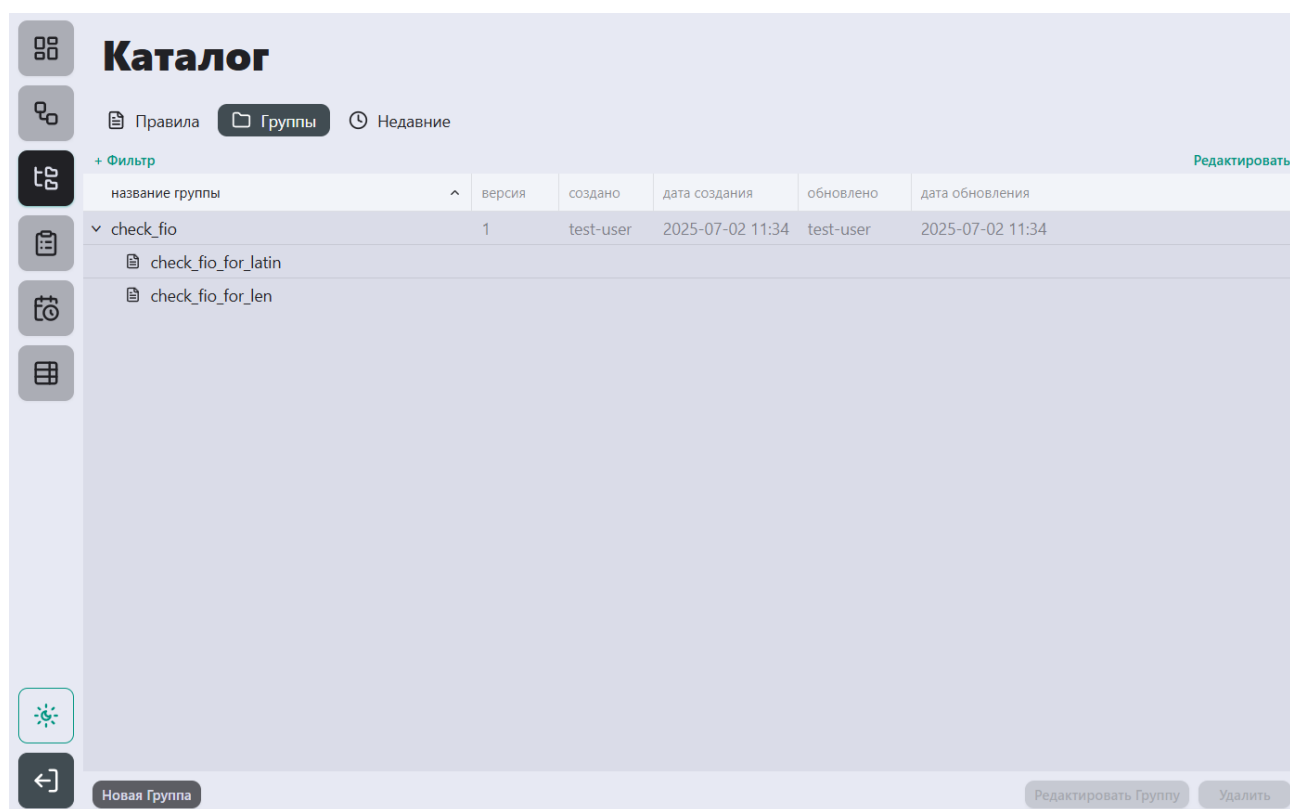
3.2 Список групп

Группа правил – это совокупность правил, объединенных вместе в соответствии с бизнес-потребностью.

Например, проверка данных физического лица, как правило, требует не одного, а нескольких правил. В этом случае создаётся группа «Проверка физ. лиц», в которую могут входить правила, такие как «Проверка СНИЛС», «Проверка ФИО» и другие.

Группы упрощают организацию каталога и позволяют удобно управлять связанными наборами правил. Они могут использоваться при создании задачи на проверку данных, однако результаты по-прежнему формируются отдельно для каждого входящего в группу правила. Таким образом, группа — это способ логически объединить правила, не влияя на их индивидуальную обработку и отчётность.

Список групп представлен в виде таблицы, аналогичной списку правил, и поддерживает сортировку и фильтрацию по ключевым параметрам (Рисунок 9). Это позволяет быстро находить нужные группы и управлять ими.



название группы	версия	создано	дата создания	обновлено	дата обновления
check_fio	1	test-user	2025-07-02 11:34	test-user	2025-07-02 11:34
check_fio_for_latins					
check_fio_for_len					

Рисунок 9 - Список групп в каталоге

3.3 Создание группы

Вы можете создать новую группу, указав список правил, которые должны в неё входить. Для этого необходимо выбрать правила из доступного каталога и присвоить группе уникальное имя (Рисунок 10).

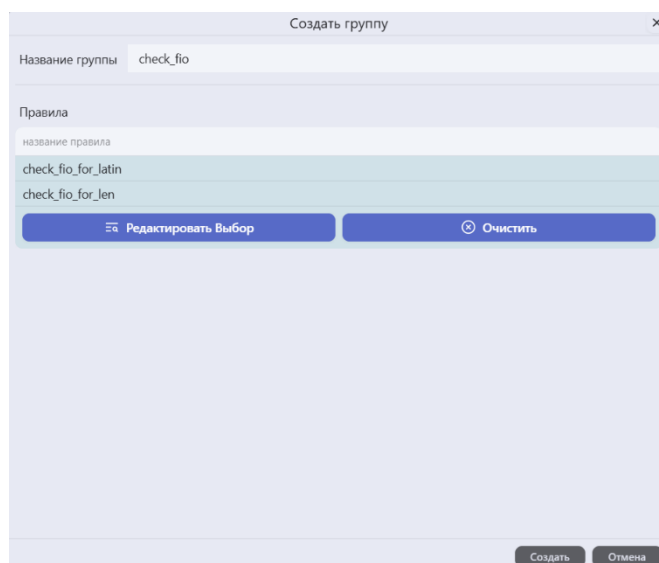


Рисунок 10 - Создание группы

Состав группы можно изменять в любое время: добавлять новые правила или удалять существующие.

4 Планирование задач

Сервис планирования задач – это сервис, который создает задачи для проверки набора данных и формирует сообщения-запросы для Сервиса исполнения правил.

Задача — настраиваемый объект, предназначенный для выполнения группы правил на заданных данных.

В рамках одной задачи инициируется создание набора проверок: каждая проверка представляет собой применение одного правила к одному объекту данных. Таким образом, одна задача может включать множество проверок, соответствующих выбранной группе правил и входным данным.

4.1 Список задач

Все задачи представлены в виде таблицы, содержащей ключевые параметры каждой задачи (Рисунок 11). Пользователь может управлять списком, применяя фильтры и сортировку по нужным полям. Это позволяет быстро находить необходимые задачи и отслеживать их статус.

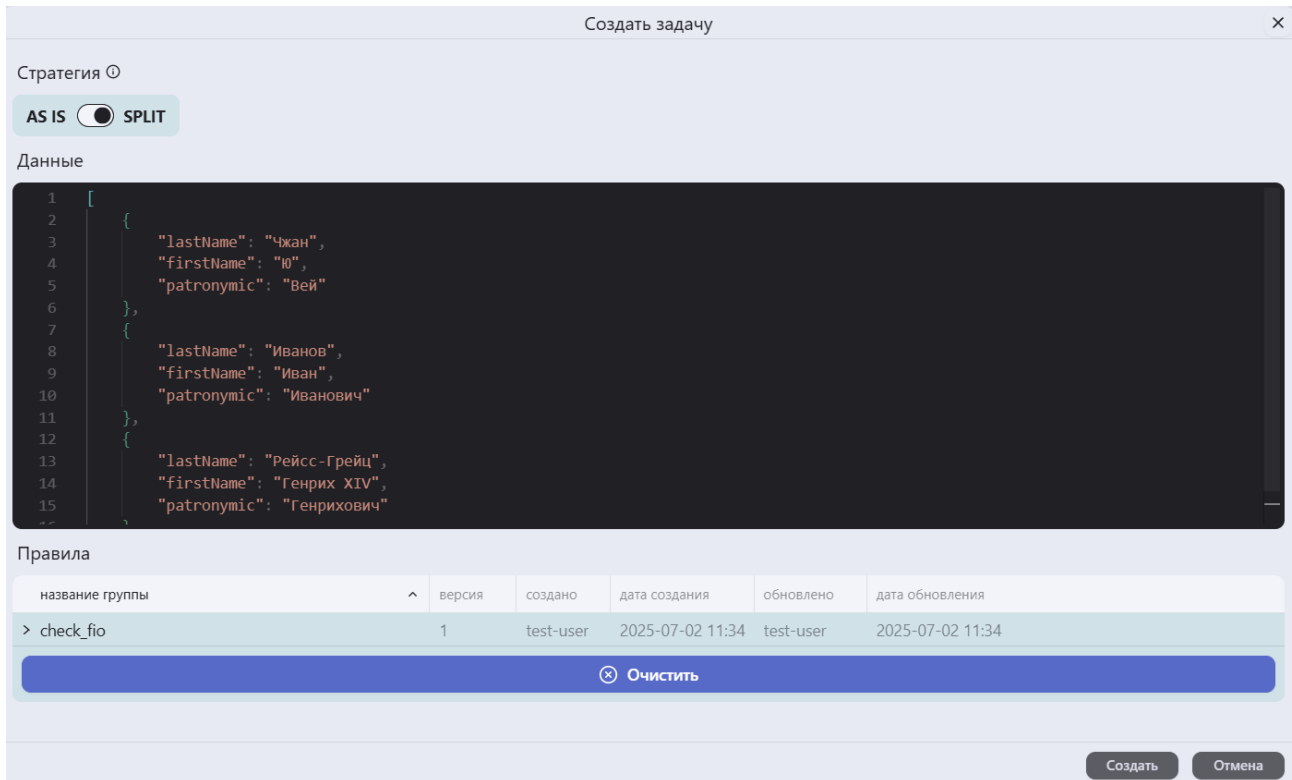
Список Задач							
+ Фильтр	id	id группы	название группы	версия	дата создания	создано	статус
	1752219360100023	1752219312100016		2	2025-07-11 10:36	dqf-airflow	отправлено
	1752219312100022	1752219312100016		1	2025-07-11 10:35	dqf-airflow	отправлено
	1752150525100021	1751445285100001	check_fio	1	2025-07-10 15:28	test-user	отправлено
	1752150252100020	1751445285100001	check_fio	1	2025-07-10 15:24	test-user	отправлено
	1752064258100019	1751447795100002		8	2025-07-09 15:30	test-user	отправлено
	1752064049100018	1751445285100001	check_fio	1	2025-07-09 15:27	test-user	отправлено
	1752063860100017	1751549198100003		2	2025-07-09 15:24	test-user	отправлено
	1752051966100016	1752051965100012		1	2025-07-09 12:06	test-user	отправлено
	1751879868100015	1751447795100002		7	2025-07-07 12:17	test-user	отправлено
	1751555472100014	1751555398100004		2	2025-07-03 18:11	test-user	отправлено
	1751549863100013	1751447795100002		6	2025-07-03 16:37	test-user	отправлено
	1751549612100012	1751445285100001	check_fio	1	2025-07-03 16:33	test-user	отправлено
	1751549198100011	1751549198100003		1	2025-07-03 16:26	test-user	отправлено
	1751535641100010	1751447795100002		5	2025-07-03 12:40	test-user	отправлено
	1751535098100009	1751445285100001	check_fio	1	2025-07-03 12:31	test-user	отправлено
	1751533593100008	1751447795100002		4	2025-07-03 12:06	test-user	отправлено
	1751533234100007	1751445285100001	check_fio	1	2025-07-03 12:00	test-user	отправлено
	1751530069100006	1751447795100002		3	2025-07-03 11:07	test-user	отправлено
	1751447965100005	1751447795100002		2	2025-07-02 12:19	test-user	отправлено
	1751447795100004	1751447795100002		1	2025-07-02 12:16	test-user	отправлено

Рисунок 11 - Список задач

4.2 Создание задачи

Пользователь может создать новую задачу со страницы со списком задач. В процессе создания необходимо указать (Рисунок 12):

- данные, которые необходимо проверить;
- стратегию разбивки данных:
 - AS IS — все переданные данные проверяются одним вызовом каждого правила в группе. Подходит для случаев, когда данные представляют собой единое значение и должны анализироваться как целое;
 - SPLIT — каждый элемент входного массива обрабатывается отдельно: правило применяется к каждому элементу по отдельности. Используется, когда данные представляют собой список однотипных значений (например, перечень физических или юридических лиц), и требуется провести индивидуальную проверку для каждого из них.
- группу правил — можно выбрать существующую группу или задать набор правил, на основе которого система сформирует внутреннюю группу.



Создать задачу

Стратегия

AS IS ☒ SPLIT ☐

Данные

```

1  [
2    {
3      "lastName": "Чжан",
4      "firstName": "Ю",
5      "patronymic": "Вей"
6    },
7    {
8      "lastName": "Иванов",
9      "firstName": "Иван",
10     "patronymic": "Иванович"
11   },
12   {
13     "lastName": "Рейсс-Грейц",
14     "firstName": "Генрих XIV",
15     "patronymic": "Генрихович"
16   }
17 ]
  
```

Правила

название группы	версия	создано	дата создания	обновлено	дата обновления
> check_fio	1	test-user	2025-07-02 11:34	test-user	2025-07-02 11:34

Очистить

Создать Отмена

Рисунок 12 - Создание задачи

4.3 Изменение статуса задачи

Пользователь может отменить задачу, но только до момента, пока она не перешла в статус «Отправлено» (Рисунок 13).

Задача может находиться в одном из следующих статусов:

- Отправлено (SENT) — все сообщения задачи переданы во входящую очередь сервиса исполнения правил;

- В очереди (QUEUED) — сообщения задачи ещё не отправлены полностью или вовсе не отправлены;
- Отменено (CANCELLED) — задача отменена пользователем до начала отправки.

Список Задач							
+ Фильтр							
id	id группы	название группы	версия	дата создания	создано	статус	
1752220175100024	1751445285100001	check_fio	1	2025-07-11 10:49	test-user	в очереди	
1752219360100023	1752219312100016		2	2025-07-11 10:36	dqf-airflow	отправлено	

Рисунок 13 - Остановка выполнения задачи

4.4 Просмотр результатов выполнения задачи

Со страницы списка задач можно открыть любую задачу и просмотреть результаты её выполнения (Рисунок 14).

Список Задач / 1752220175100024

Задача 1752220175100024

SENT

Создать отчет test-user 2025-07-11 10:49

Задача завершена

6 запросов из 6

Результаты

+ Фильтр

название правила	статус	получено
check_fio_for_len	не пройдено	2025-07-11 10:50
<pre>{ "input": { "lastName": "Чхан", "firstName": "И", "patronymic": "Бей" }, "errorDescription": "Фамилия, имя или отчество содержат меньше 2 символов" }</pre>		
check_fio_for_len	пройдено	2025-07-11 10:50
<pre>{ "input": { "lastName": "Рейсс-Грейц", "firstName": "Генрих XIV", "patronymic": "Генрихович" } }</pre>		
check_fio_for_len	пройдено	2025-07-11 10:50
check_fio_for_latin	не пройдено	2025-07-11 10:50
check_fio_for_latin	пройдено	2025-07-11 10:50

Рисунок 14 - Результаты выполнения задачи

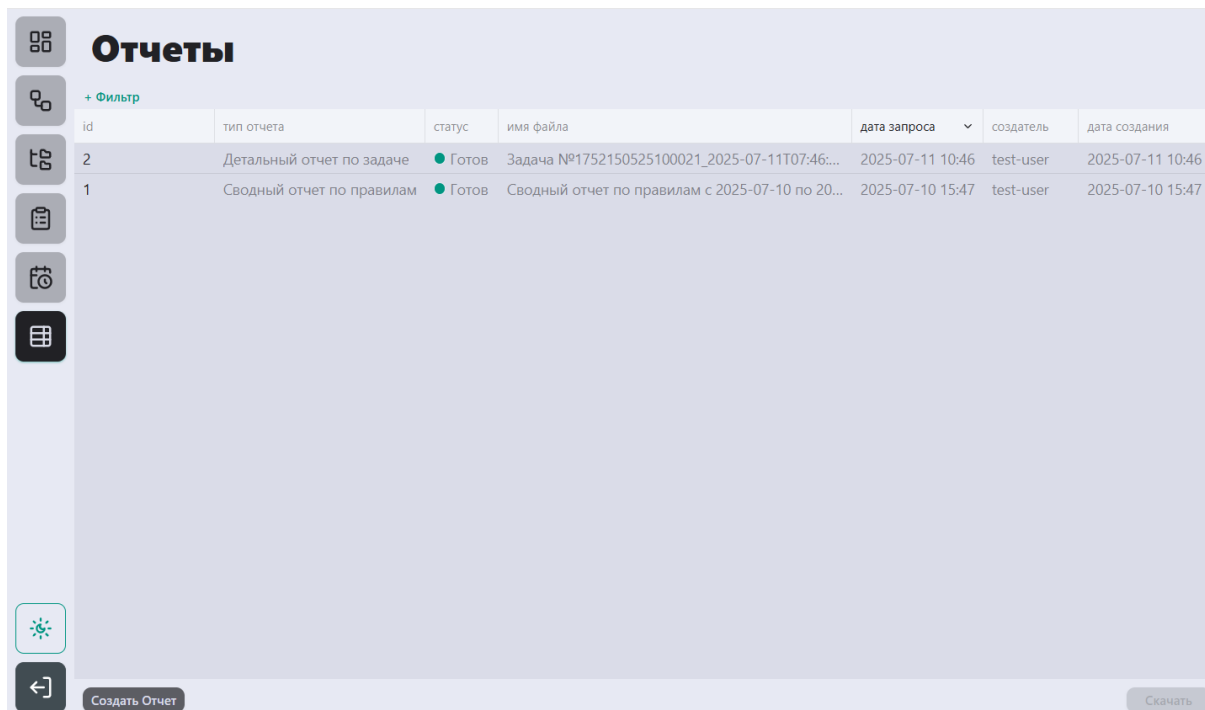
Результаты отображаются по каждому вызову правила отдельно. Для каждого вызова указаны:

- Статус выполнения:
 - Пройдено — ошибок в проверяемых данных не обнаружено;
 - Не пройдено — выявлены ошибки в данных;
 - Ошибка — при выполнении правила возникла техническая ошибка.

- Дополнительная информация, возвращаемая в результате выполнения правила. Она представлена в формате JSON и может содержать произвольные поля, заданные при конфигурации правила. *Например, при нарушении формата может быть возвращён код качества: "code": "INVALID_FORMAT".*

5 Отчетность

Для анализа результатов проверок пользователю доступны встроенные инструменты отчётности (Рисунок 15).



Отчеты							
+ Фильтр							
id	тип отчета	статус	имя файла	дата запроса	создатель	дата создания	
2	Детальный отчет по задаче	Готов	Задача №1752150525100021_2025-07-11T07:46:...	2025-07-11 10:46	test-user	2025-07-11 10:46	
1	Сводный отчет по правилам	Готов	Сводный отчет по правилам с 2025-07-10 по 20...	2025-07-10 15:47	test-user	2025-07-10 15:47	

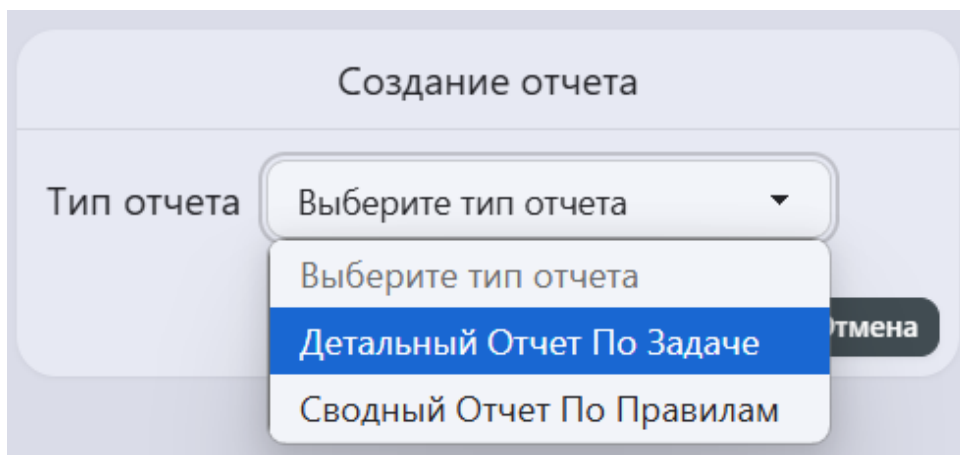
Создать Отчет

Скачать

Рисунок 15 - Запросы на создание отчетов

Система позволяет сформировать один из следующих видов отчётов (Рисунок 16):

- Детальный отчёт по задаче — содержит полную информацию о каждом вызове правила внутри выбранной задачи. Этот отчёт практически полностью повторяет структуру страницы просмотра задачи и может использоваться для выгрузки или архивации данных.
- Сводный отчёт по правилам — включает статистику выполнения каждого правила за выбранный период времени: количество проверок со статусами «пройдено», «не пройдено» и «ошибка». Используется для общего анализа качества данных и выявления проблемных участков.



Создание отчета

Тип отчета

Выберите тип отчета

Выберите тип отчета

Детальный Отчет По Задаче

Сводный Отчет По Правилам

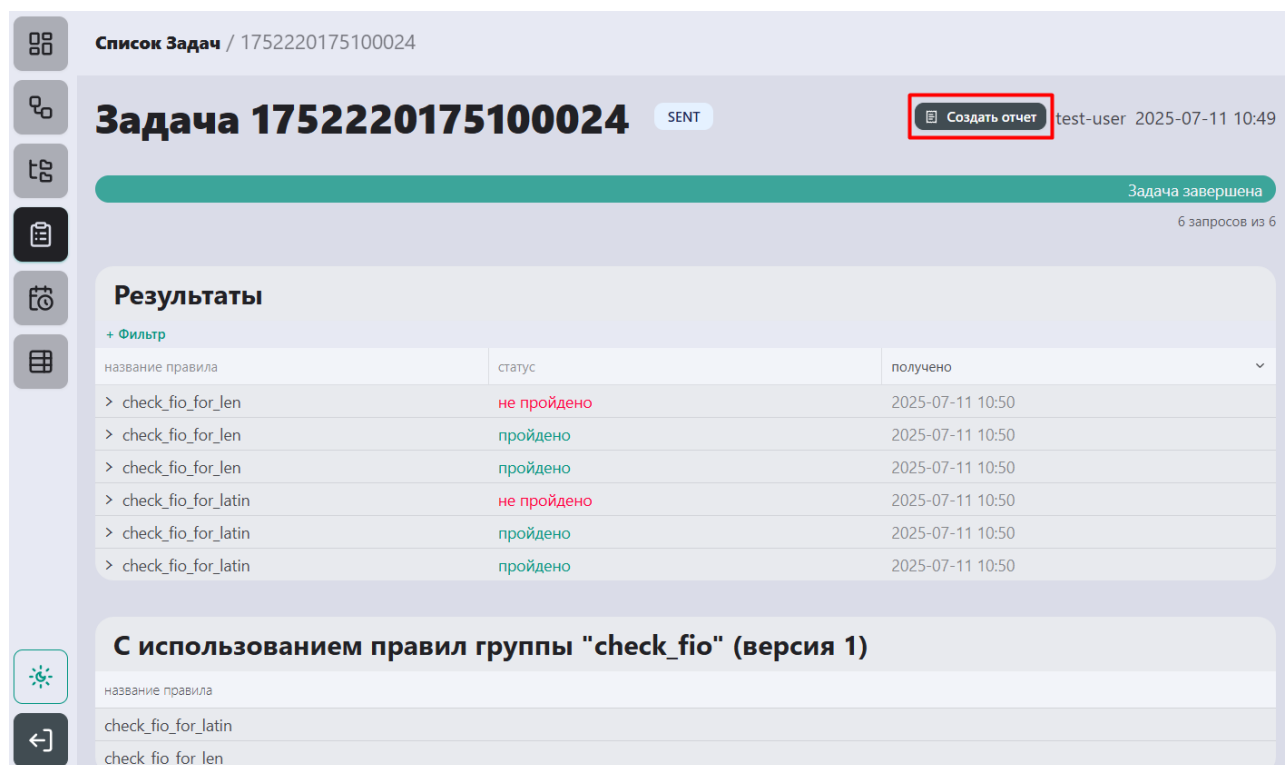
Отмена

Рисунок 16 - Выбор типа отчета

5.1 Формирование и загрузка отчетов

Создание отчёта начинается с запроса на его формирование. Это можно сделать:

- со страницы со всеми запросами на отчёты — для обоих типов отчётов;
- напрямую со страницы задачи — для создания детального отчёта по конкретной задаче (Рисунок 17).



Список Задач / 1752220175100024

Задача 1752220175100024 SENT Создать отчет test-user 2025-07-11 10:49

Задача завершена
6 запросов из 6

Результаты

+ Фильтр

название правила	статус	получено
> check_fio_for_len	не пройдено	2025-07-11 10:50
> check_fio_for_len	пройдено	2025-07-11 10:50
> check_fio_for_len	пройдено	2025-07-11 10:50
> check_fio_for_len	пройдено	2025-07-11 10:50
> check_fio_for_lat	не пройдено	2025-07-11 10:50
> check_fio_for_lat	пройдено	2025-07-11 10:50
> check_fio_for_lat	пройдено	2025-07-11 10:50

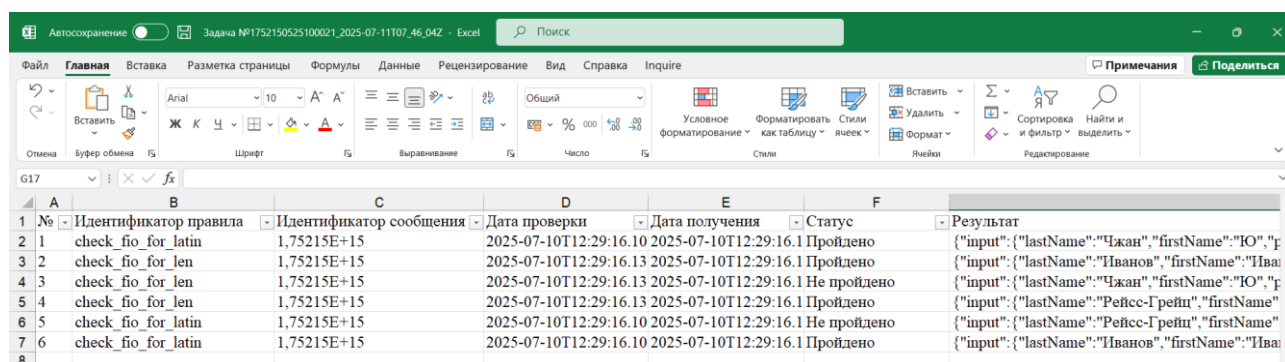
С использованием правил группы "check_fio" (версия 1)

название правила
check_fio_for_lat
check_fio_for_len

Рисунок 17 - Создание отчета со страницы задачи

После завершения формирования отчёт становится доступен для загрузки.

Все отчёты сохраняются в формате .xlsx и могут быть загружены повторно в любой момент со страницы с отчётами. Ниже представлен пример отчета типа «Детальный отчет по задаче» (Рисунок 18).



№	Идентификатор правила	Идентификатор сообщения	Дата проверки	Дата получения	Статус	Результат
1	check_fio_for_lat	1,75215E+15	2025-07-10T12:29:16.10	2025-07-10T12:29:16.1	Пройдено	{ "input": { "lastName": "Иванов", "firstName": "Иван" }, "result": "Пройдено" }
2	check_fio_for_len	1,75215E+15	2025-07-10T12:29:16.13	2025-07-10T12:29:16.1	Пройдено	{ "input": { "lastName": "Иванов", "firstName": "Иван" }, "result": "Пройдено" }
3	check_fio_for_len	1,75215E+15	2025-07-10T12:29:16.13	2025-07-10T12:29:16.1	Не пройдено	{ "input": { "lastName": "Иванов", "firstName": "Иван" }, "result": "Не пройдено" }
4	check_fio_for_len	1,75215E+15	2025-07-10T12:29:16.13	2025-07-10T12:29:16.1	Пройдено	{ "input": { "lastName": "Иванов", "firstName": "Иван" }, "result": "Пройдено" }
5	check_fio_for_lat	1,75215E+15	2025-07-10T12:29:16.10	2025-07-10T12:29:16.1	Не пройдено	{ "input": { "lastName": "Иванов", "firstName": "Иван" }, "result": "Не пройдено" }
6	check_fio_for_lat	1,75215E+15	2025-07-10T12:29:16.10	2025-07-10T12:29:16.1	Пройдено	{ "input": { "lastName": "Иванов", "firstName": "Иван" }, "result": "Пройдено" }

Рисунок 18 – Пример детального отчета по задаче

6 Дашборды

Для оценки результатов проверок и оперативного анализа качества данных в системе предусмотрена возможность создания дашбордов — интерактивных панелей с графиками и ключевыми метриками (Рисунок 19).

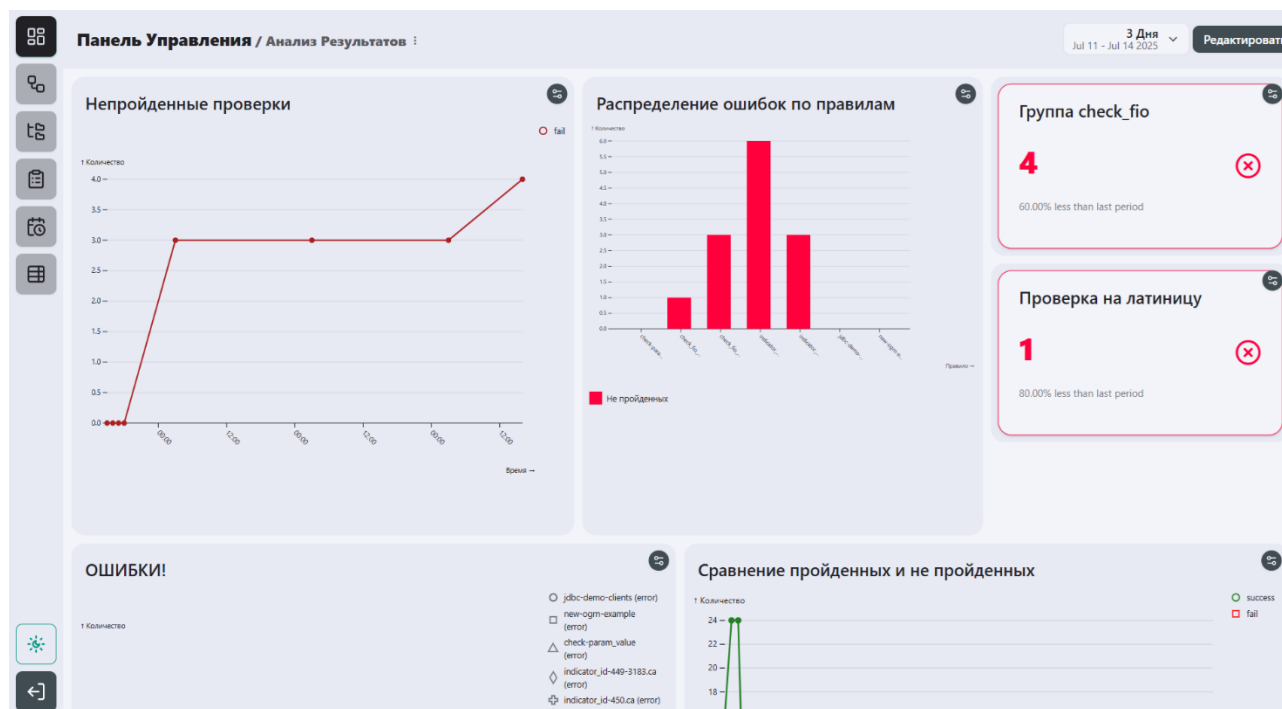


Рисунок 19 – Пример дашборда

Дашборды позволяют пользователям разных ролей — от технических специалистов до бизнес-аналитиков — отслеживать состояние и динамику проверок данных в удобном визуальном формате.

- Для аналитиков это — способ контролировать качество информации, выявлять проблемные участки и принимать решения на основе статистики.
- Для технических команд — инструмент наблюдения за стабильностью исполнения правил и возможностью быстро реагировать на возникающие ошибки.

Каждый пользователь может создать собственную панель с нужными графиками и показателями, адаптированной под его задачи. Гибкая настройка фильтров, уровней детализации и визуального представления делает дашборды мощным инструментом ежедневного контроля и анализа.

В приложении предусмотрена возможность создания и управления пользовательскими панелями дашбордов.

Пользователь может (Рисунок 20):

- создать новую панель управления;
- открыть из существующих;
- удалить любую из доступных панелей.

Открыть					
+ Фильтр					
ID	Имя	Дата создания	Создано	Дата обновления	Обновлено
1752237346100004	Demo	2025-07-11 15:35	test-user	2025-07-11 15:36	test-user
1752227503100003	Новая панель	2025-07-11 12:51	test-user	2025-07-11 12:51	test-user
1752222797100002	Анализ результатов	2025-07-11 11:33	test-user	2025-07-11 13:31	test-user
1751546505100001	Демо	2025-07-03 15:41	test-user	2025-07-10 15:45	test-user
1751533887100000	Дашборд	2025-07-03 12:11	test-user	2025-07-03 12:11	test-user

Создать Панель Управления
Открыть

Рисунок 20 - Управление дашбордами

6.1 Добавление и изменение виджетов

На панель можно добавить один из трёх типов виджетов:

- Линейная диаграмма — отображает изменение показателей во времени (Рисунок 21);
- Столбчатая диаграмма — показывает распределение проверок по выбранному параметру (Рисунок 22);
- Карточка — компактный индикатор с численным значением.

Настройки графиков (линейного и столбчатого):

- Имя виджета — произвольное название, задаваемое пользователем;
- Параметр по оси X — для *линейного* графика доступна только дата/время, для *столбчатого* — распределение по правилам, группам или расписаниям;
- Параметр по оси Y — возможен выбор одного или нескольких значений: количество пройденных, не пройденных проверок и ошибок;
- Иерархия — настройка drill-down: можно указать начальный уровень (расписания, группы, правила);

- Фильтр — по нужным правилам, группам или расписаниям.



Рисунок 21 – Линейная диаграмма

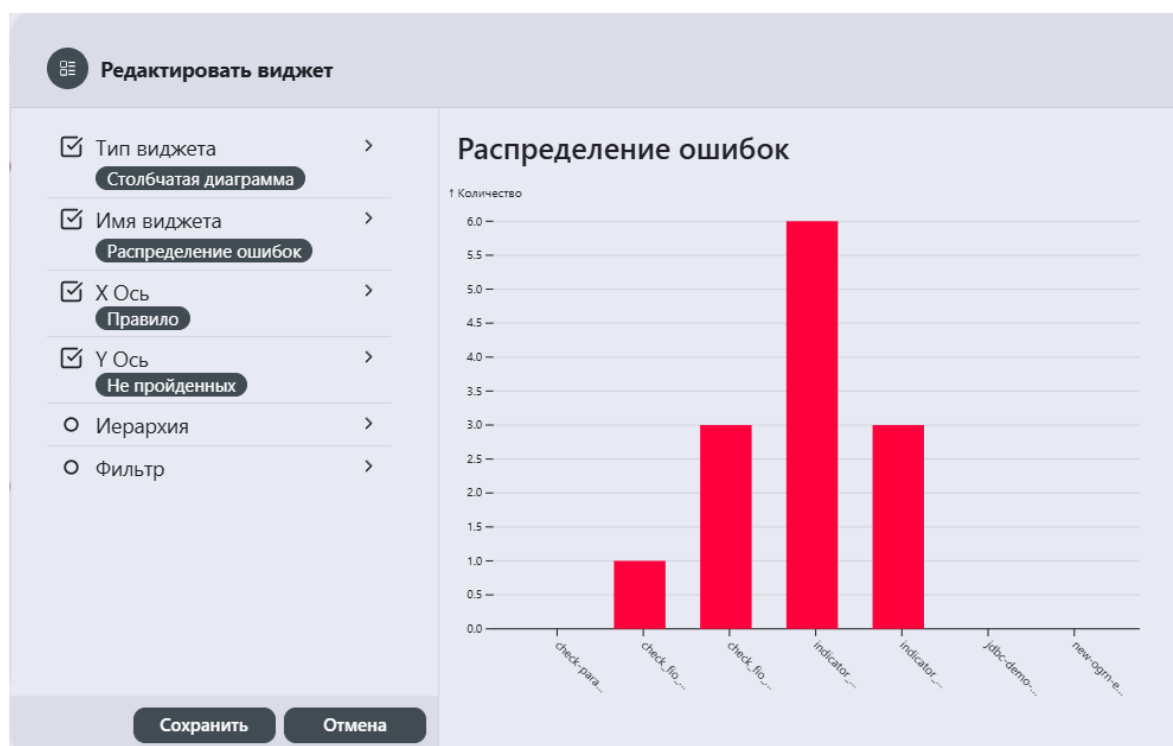


Рисунок 22 – Столбчатая диаграмма

Настройки карточки (Рисунок 23):

- Имя виджета;
- Показатель — один из трёх: количество пройденных, не пройденных проверок или ошибок;
- Фильтр — по выбранным правилам, группам или расписаниям.

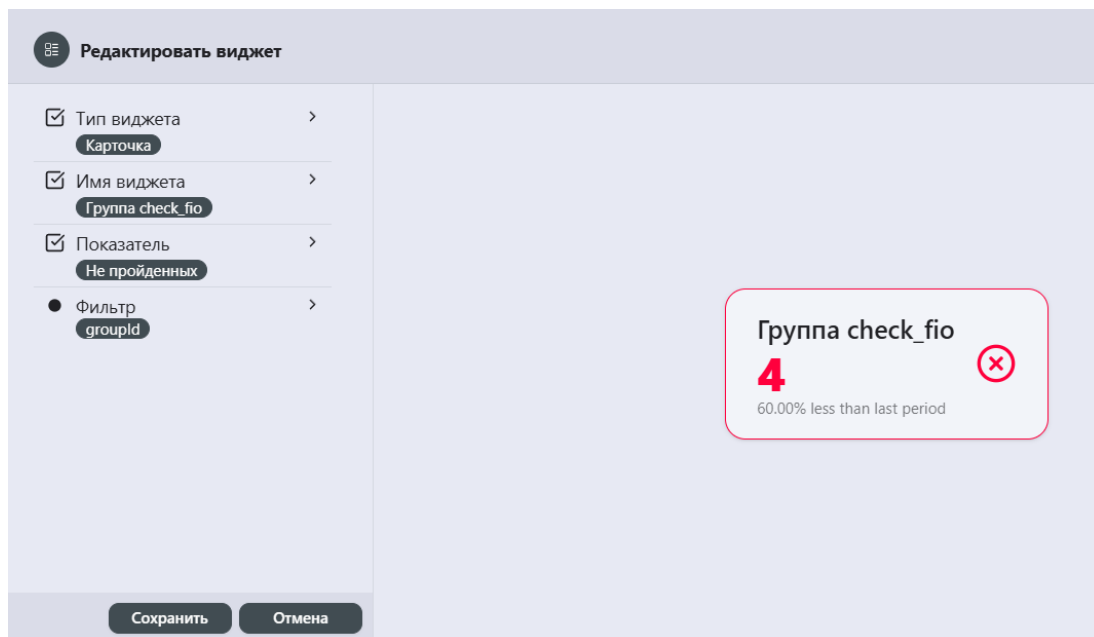


Рисунок 23 - Карточка

6.2 Редактирование дашборда

Редактирование панели включает:

- изменение расположения виджетов (drag-and-drop) (Рисунок 24),
- задание общего временного интервала, на основании которого строятся все графики панели,
- изменение имени панели.

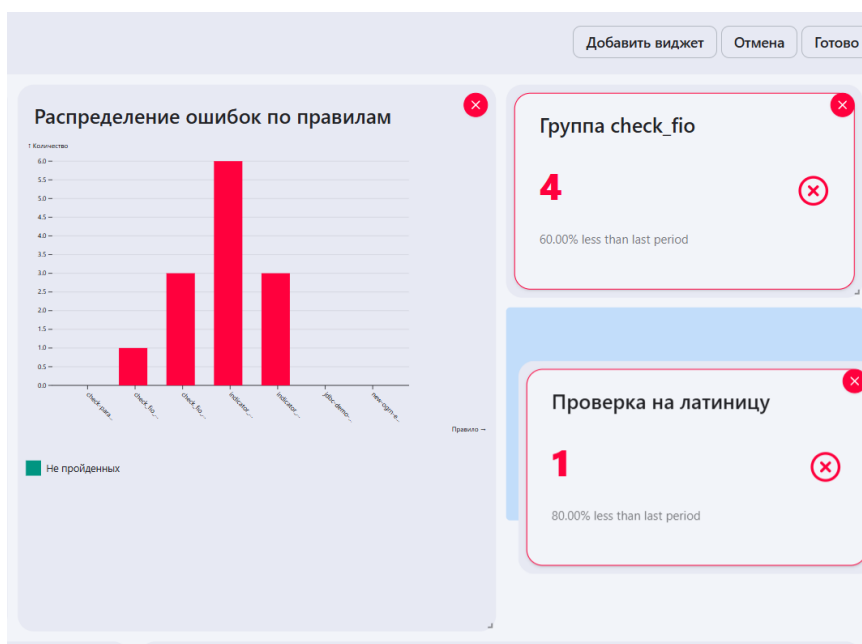


Рисунок 24 - Перемещение виджета на дашборде