

**ИНФОРМАЦИЯ, НЕОБХОДИМАЯ ДЛЯ ЭКСПЛУАТАЦИИ  
ЭКЗЕМПЛЯРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ  
METASFERA ANALYTICS**

г. Москва

2025 г.

## Оглавление

<b>1. НАЗНАЧЕНИЕ ДОКУМЕНТА .....</b>	<b>3</b>
<b>2. ПОРЯДОК РАБОТЫ С ПО .....</b>	<b>3</b>
2.1. Ролевая модель .....	3
2.2. Общая информация по ПО, работа в ПО и общие элементы интерфейса.....	3
2.2.1. Назначение ПО .....	3
2.2.2. Основные функции ПО .....	3
2.3. Руководство пользователя по работе с ПО .....	3
2.3.1. Авторизация пользователя в системе .....	3
2.3.2. Главный экран: как перейти в ПО .....	4
2.3.3. Просмотр отчетов .....	5
2.4. Разработка отчетов .....	10
2.4.1. Подготовка рабочего пространства.....	10
2.4.2. Создание скриптов заполнения и отправки конфигураций на сервер РМС ВІ .....	11
2.4.3. Описание ETL-конфигурации.....	12
2.4.4. Описание OLAP-конфигурации и TABLE-конфигурации .....	13
2.4.5. Описание DASHBOARD- конфигурации.....	14
2.4.6. Описание элементов DASHBOARD- конфигурации .....	15
2.4.7. Описание основного файла запуска скриптов .....	16
2.4.8. Запуск скрипта создания дашбордов .....	17
2.4.9. Открытие созданных дашбордов.....	18
2.4.10.Просмотр кубов.....	19
2.4.11.Просмотр настроек ETL .....	19
2.4.12.Просмотр процессов ETL .....	20
2.4.13.Добавление нового отчета.....	21
2.4.14.Экспорт и импорт отчета.....	21

## 1. Назначение документа

Данный документ служит руководством пользователя по эксплуатации программного обеспечения Metasfera Analytics (ПО, Система).

## 2. Порядок работы с ПО

### 2.1. Ролевая модель

Список ролей пользователей ПО и их основные функции представлены в Таблице 1.

Таблица 1 - Ролевая модель

Наименование роли	Основные функции роли
Администратор системы	Организационная единица, имеющая полные административные права к системе
Пользователь системы	Организационная единица, выполняющая работы с системой, вносящая информацию, но не имеющая возможности корректировать права других пользователей

### 2.2. Общая информация по ПО, работа в ПО и общие элементы интерфейса

#### 2.2.1. Назначение ПО

ПО предназначено для обеспечения пользователей аналитической информацией о ходе реализации проектов капитального строительства на всех этапах их жизненного цикла.

#### 2.2.2. Основные функции ПО

Основными функциями ПО являются:

- Формирование отчетов в соответствии с настроенными формами отчетов;
- Получение данных из смежных модулей системы.

ПО поддерживает многопользовательский режим работы.

## 2.3. Руководство пользователя по работе с ПО

### 2.3.1. Авторизация пользователя в системе

На странице авторизации требуется ввести имя пользователя и пароль:

- в поле Пользователь требуется ввести уникальное имя пользователя (Например, user);
- в поле Пароль требуется ввести персональный пароль пользователя (Например, password);

далее необходимо нажать кнопку «Войти» для подключения (Рисунок 1).



**Рисунок 1. Авторизация пользователя в системе**

***Примечание!***

Поля Пользователь и Пароль заполняются с учетом регистра.

### **2.3.2. Главный экран: как перейти в ПО**

После авторизации пользователю необходимо выбрать продукт Metasfera Analytics (Рисунок 2).

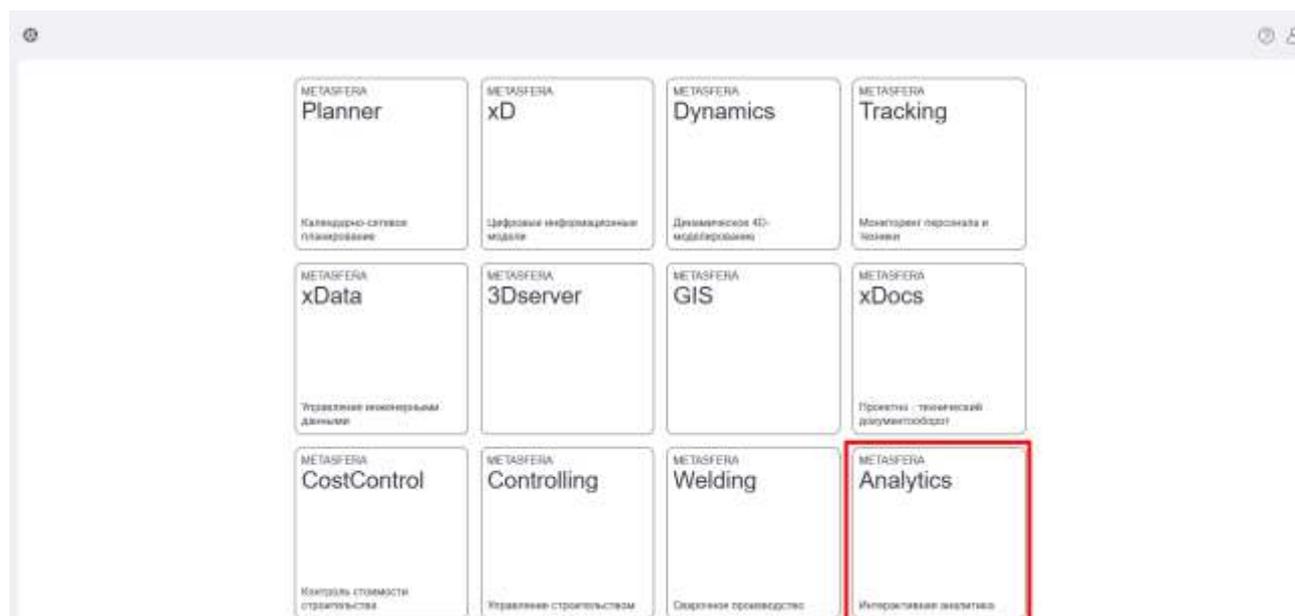


Рисунок 2. Переход в Metasfera Analytics

Главный экран ПО содержит навигационное меню внизу экрана.



Рисунок 3. Навигационное меню

### 2.3.3. Просмотр отчетов

#### 2.3.3.1. Просмотр отчета «Реестр проектов»

После перехода в ПО открывается отчет «Реестр проектов».

Реестр имеет вид таблицы. Пример реестра проектов приведен на рисунке ниже (Рисунок 4).

Имя	Ц/У	Адрес	Ц/У	Исполнение		Объем работ			Уровень выполнения			Уровень выполнения					
				нач. Ц/У	оконч. Ц/У	нач. Ц/У	оконч. Ц/У	факт	план Ц/У	оконч. Ц/У	факт	план Ц/У	оконч. Ц/У	факт	план Ц/У	оконч. Ц/У	
Проект	0000	0000	0000	00.01.2025	00.01.2025	0	0	0	0	0	0	0	0	0	0	0	0
Проект	0000	0000	0000	19.07.2024	19.07.2024	0	0	0	0	0	0	0	0	0	0	0	0
Проект	0000	0000	0000	00.12.2024	00.04.2025	0	0	0	0	0	0	0	0	0	0	0	0
Проект	0000	0000	0000	00.11.2024	00.12.2024	0	0	0	0	0	0	0	0	0	0	0	0
Проект	0000	0000	0000	00.10.2024	00.11.2024	0	0	0	0	0	0	0	0	0	0	0	0

Рисунок 4. Отчет «Реестр проектов»

### 2.3.3.2. Просмотр отчета «СВОД»

Для просмотра отчета «СВОД» необходимо в реестре проектов выбрать требуемый проект и после этого в нижнем навигационном меню выбрать отчет «СВОД» (Рисунок 5).

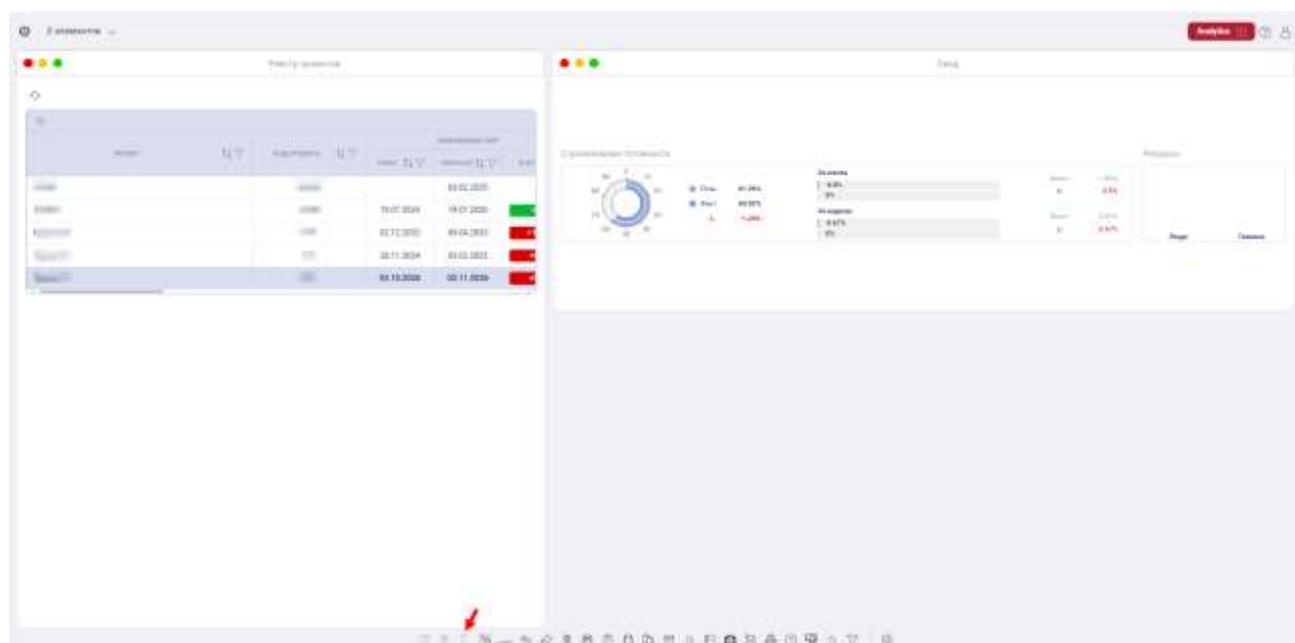


Рисунок 5. Отчет «СВОД»

### 2.3.3.3. Просмотр отчета «S-кривая строительной готовности»

Для просмотра отчета «S-кривая строительной готовности» необходимо в реестре проектов выбрать требуемый проект и после этого в нижнем навигационном меню выбрать отчет «S-кривая строительной готовности» (Рисунок 6).

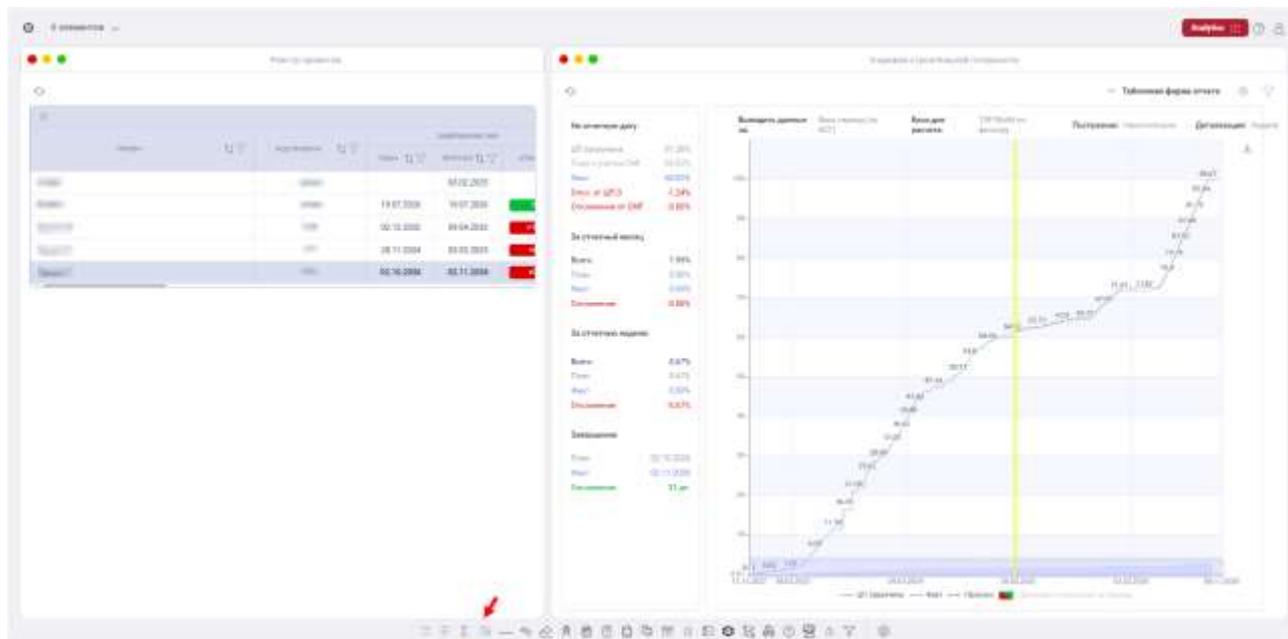


Рисунок 6. Отчет «S-кривая строительной готовности»

### 2.3.3.4. Просмотр отчета «Ключевые вехи»

Для просмотра отчета «Ключевые вехи» необходимо в реестре проектов выбрать требуемый проект и после этого в нижнем навигационном меню выбрать отчет «Ключевые вехи» (Рисунок 7).

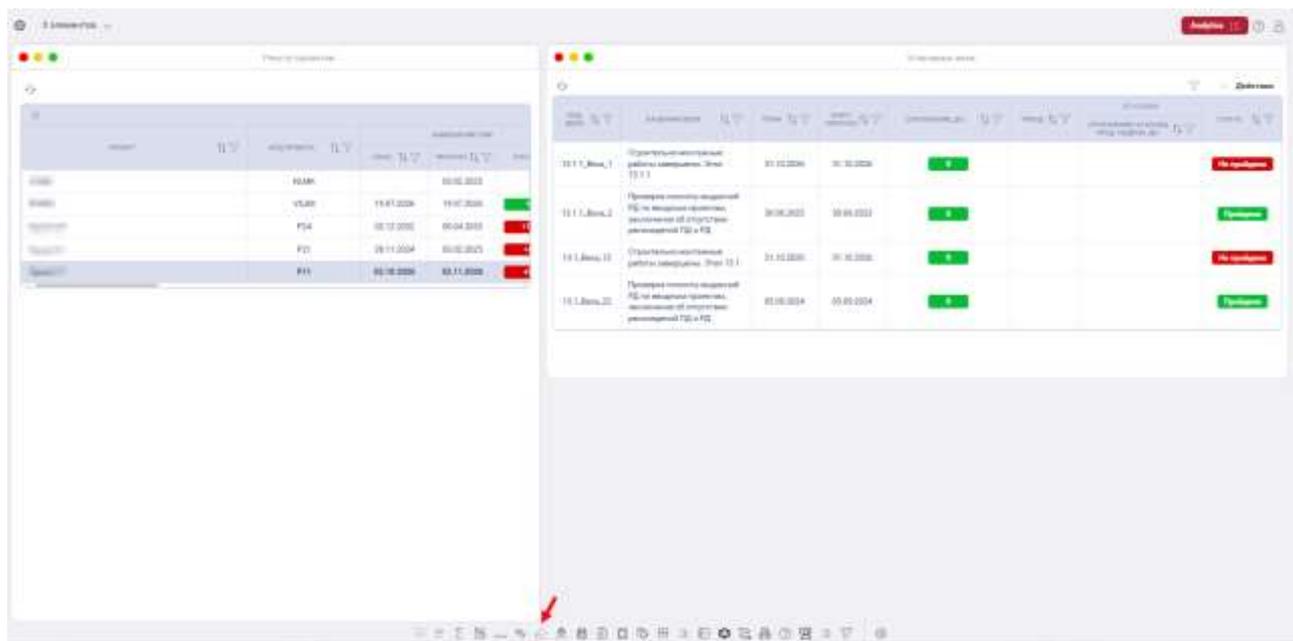


Рисунок 7. Отчет «Ключевые вехи»

### 2.3.3.5. Просмотр отчета «Ресурсы»

Для просмотра отчета «Ресурсы» необходимо в реестре проектов выбрать требуемый проект» и после этого в нижнем навигационном меню выбрать отчет «Ресурсы (Рисунок 8).

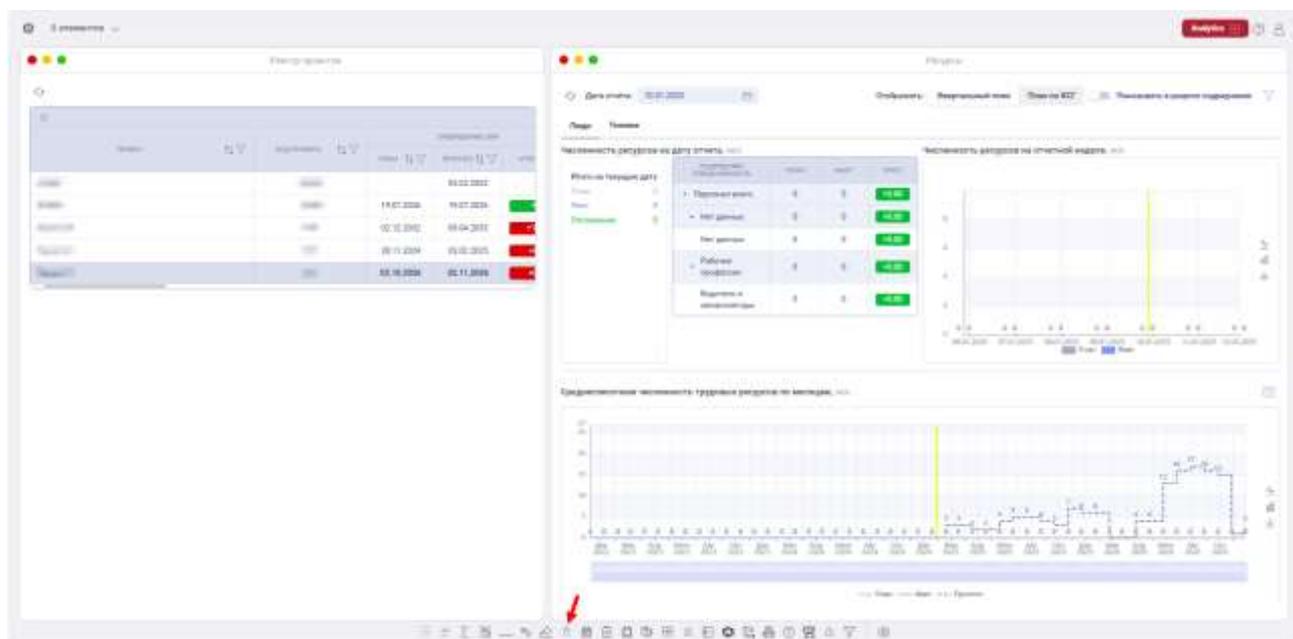


Рисунок 8. Отчет «Ресурсы»

### 2.3.3.6. Просмотр отчета «Статус МТО в разрезе этапов поставки»

Для просмотра отчета «Статус МТО в разрезе этапов поставки» необходимо в реестре проектов выбрать требуемый проект и после этого в нижнем навигационном меню выбрать отчет «Статус МТО в разрезе этапов поставки» (Рисунок 9).

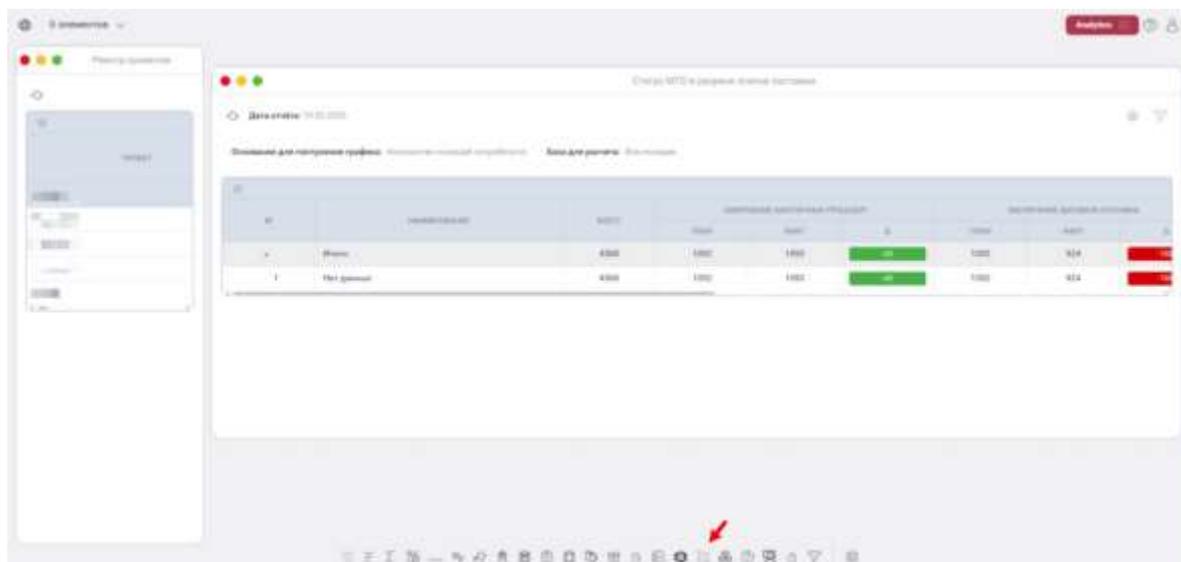


Рисунок 9. Отчет «Статус МТО в разрезе этапов поставки»

### 2.3.3.7. Просмотр отчета «Открытые вопросы»

Для просмотра отчета «Открытые вопросы» необходимо в реестре проектов выбрать требуемый проект и после этого в нижнем навигационном меню выбрать отчет «Открытые вопросы» (Рисунок 10).



Рисунок 10. Отчет «Открытые вопросы»

### 2.3.3.8. Просмотр отчета «Исполнение поручений»

Для просмотра отчета «Исполнение поручений» по проекту необходимо в реестре проектов выбрать требуемый проект и в навигационном меню выбрать отчет «Исполнение поручений» (Рисунок 11).

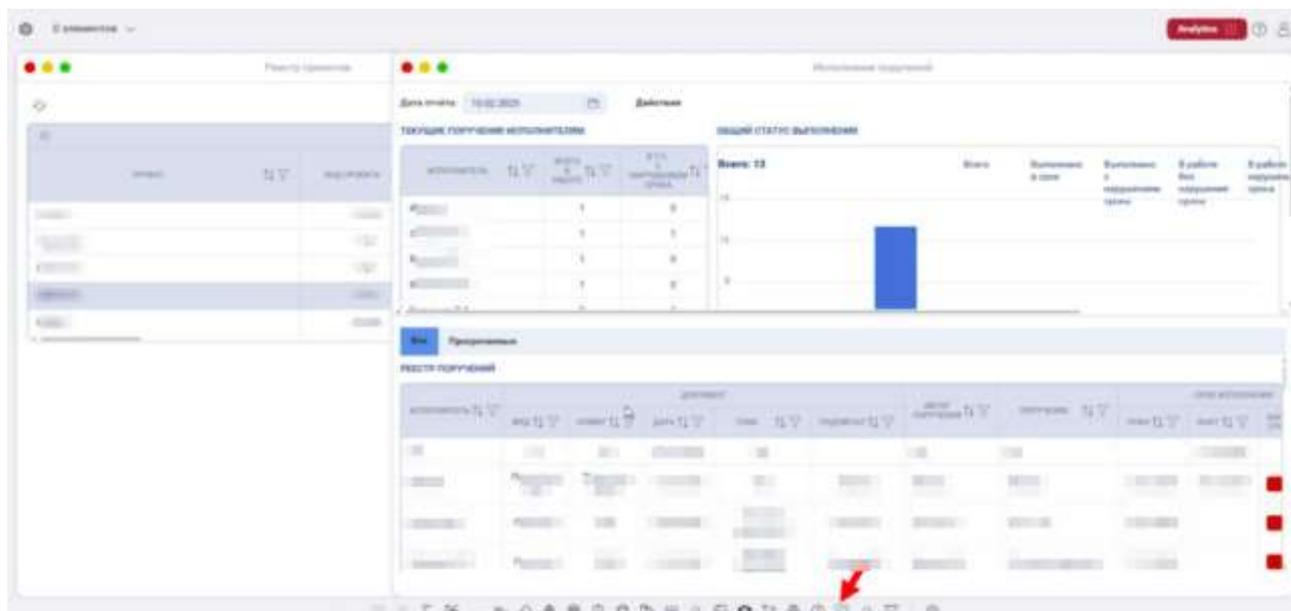


Рисунок 11. Отчет «Исполнение поручений»

## 2.4. Разработка отчетов

### 2.4.1. Подготовка рабочего пространства

Необходимо установить Node.js последней версии из 16.x.x, скачав установочный файл по ссылке <https://nodejs.org/en/about/previous-releases> для своей операционной системы.

Далее необходимо скачать zip-архив с утилитой создания дашбордов `dashboar-creator-xxxx-xx-xx.zip`. Распаковать его содержимое в папку с наименованием этого архива.

В терминале перейти в папку с распакованным содержимым:

```
cd dashboar-creator-xxxx-xx-xx/
```

Выполнить команду:

```
npm install
```

Эта команда установит необходимые зависимости в `dashboar-creator-xxxx-xx-xx/node_modules/`.

## 2.4.2. Создание скриптов заполнения и отправки конфигураций на сервер РМС VI

Перейти в папку `dashboar-creator-xxxx-xx-xx/examples/`.

Создать в ней папку с наименованием проекта в рамках, которого создаются скрипты.

Перейти в эту папку `dashboar-creator-xxxx-xx-xx/examples/name_some_project/`.

Создать в этой папке папку с наименованием схемы `dashboar-creator-xxxx-xx-xx/examples/name_some_project/name_some_schema/`. Именно с наименованием схемы создавать необязательно, просто это удобно для структурирования информации.

Схема – условная объединяющая сущность, содержащая все конфигурации и настройки дашбордов в рамках одного источника ETL. В дальнейшем она участвует в формировании пути в URL-адресе дашборда и фигурирует в скриптах, как `schemaId`. Из названия схемы должно быть понятно, какого рода информацию содержат дашборды внутри неё.

Внутри папки с наименованием схемы создаём файлы и папки по аналогии с рекомендованной структурой в папке `dashboar-creator-xxxx-xx-xx/examples/base/`.

Примерная рекомендованная структура папок и файлов представлена на Рисунок 12.

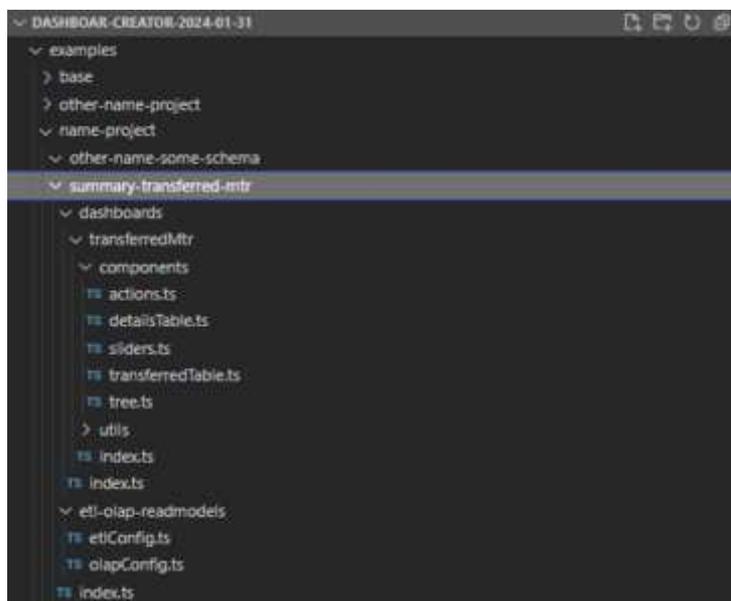


Рисунок 12. Примерная структура папок и файлов для схемы

### 2.4.3. Описание ETL-конфигурации

Для создания ETL-конфигурации необходимо указать расписание cron в пятизначном формате, источник данных с описанием параметров подключения к базе данных, последовательные операции ETL-процесса (jobs - шаги, которые будут выполнены при получении данных из источника, их преобразовании и загрузке в базу назначения).

Пример заполнения ETL-конфигурации представлен на Рисунок 13.

```
1 import { ETLConfig } from "@pmc-bi/dashboard-creator";
2
3 export const etlConfig: ETLConfig = {
4   cron: "0 0 15 7 *",
5   sources: {
6     postgresql: {
7       type: "PGSQL",
8       payload: {
9         user: "db_user_name",
10        password: "db_user_pass",
11        database: "db_name",
12        host: "db_ip_host",
13        port: 5432,
14      },
15    },
16  },
17  jobs: [
18    {
19      type: "PGSQL.EXTRACT_FROM_CUSTOM_EXECUTE_QUERY",
20      payload: {
21        sourceName: "pgsql",
22        destinationTableName: "mtr_table",
23        sql: `
24          SELECT * FROM dev.db_table_name
25        `,
26      },
27    },
28    {
29      type: "Warehouse.EXECUTE_SQL",
30      payload: {
31        sql: `
32          INSERT INTO $(olapSchemaName~).mtr_table
33            (id, project_id, stage_name, object_name, quantity)
34          SELECT
35            ROW_NUMBER() OVER () AS id, project_id, stage_name, object_name, quantity
36          FROM $(etlSchemaName~).mtr_table
37        `,
38      },
39    },
40  ],
41 };
```

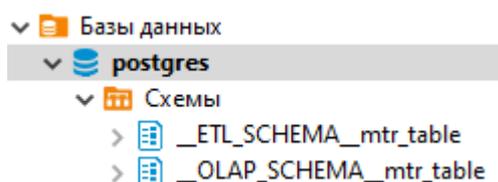
Рисунок 13. Пример заполнения ETL-конфигурации

При выполнении операции «EXTRACT\_FROM\_CUSTOM\_EXECUTE\_QUERY» данные извлекаются из источника и помещаются в таблицу destinationTableName внутри схемы БД с наименованием \_\_ETL\_SCHEMA\_\_name\_schema.

Операция с типом «Warehouse.EXECUTE\_SQL» используется для преобразования данных. При выполнении этой операции данные выбираются из таблицы в схеме БД \_\_ETL\_SCHEMA\_\_schema\_name, преобразуются внутри SELECT и помещаются в таблицу в схеме БД \_\_OLAP\_SCHEMA\_\_schema\_name в указанные в INSERT-е поля.

Параметры \$(etlSchemaName~), \$(olapSchemaName~) – это заполнители, которые используются для динамического формирования наименования схем БД на основе наименования схемы дашбордов.

Промежуточные схемы БД, формируемые в результате выполнения процесса ETL представлены на Рисунок 14.



**Рисунок 14. Промежуточные схемы БД при выполнении ETL-процесса**

#### **2.4.4. Описание OLAP-конфигурации и TABLE-конфигурации**

Эти конфигурации предназначены для описания таблиц назначения, в которые будут загружены данные в результате выполнения процесса ETL для формирования на их основе дашбордов.

OLAP-конфигурация содержит в своём описании условное деление полей на измерения и факты. TABLE-конфигурация содержит просто описание полей таблицы, создаваемой в базе назначения.

Примеры заполнения OLAP-конфигурации и TABLE-конфигурации представлены на Рисунок 15 и Рисунок 16.

```
1 import { OlapConfig, TableConfig } from "@pmc-bi/dashboard-creator";
2
3 export const olapConfig: OlapConfig = {
4   primaryKeyName: "id",
5   primaryKeyType: "bigint",
6   dimensionKeyPostfix: "_id",
7   dimensions: [
8     {
9       tableName: "mtr_table",
10      parentTablesNames: [],
11      fields: [
12        { dimensionName: "project_id", fieldName: "project_id", type: "integer" },
13        { dimensionName: "stage_name", fieldName: "stage_name", type: "varchar" },
14        { dimensionName: "object_name", fieldName: "object_name", type: "varchar" },
15      ],
16    },
17  ],
18  facts: [
19    {
20      tableName: "mtr_table",
21      fields: [
22        { fieldName: "quantity", type: "double" },
23      ],
24    },
25  ],
26 };
```

Рисунок 15. Пример заполнения OLAP-конфигурации

```
1 import { OlapConfig, TableConfig } from "@pmc-bi/dashboard-creator";
2
3 export const tableConfigs: TableConfig[] = [
4   {
5     name: "mtr_table",
6     columns: [
7       { name: "id", type: "bigint" },
8       { name: "project_id", type: "integer" },
9       { name: "construction_name", type: "varchar" },
10      { name: "stage_name", type: "varchar" },
11      { name: "object_name", type: "varchar" },
12      { name: "quantity", type: "double" },
13    ],
14    primaryKey: "id",
15  },
16 ];
```

Рисунок 16. Пример заполнения TABLE- конфигурации

#### 2.4.5. Описание DASHBOARD- конфигурации

Эта конфигурация предназначена для описания визуальной части дашборда. Она содержит в себе наименование дашборда, наименования и описание виджетов, описание размещения элементов дашборда, обработчики различных событий.

Пример заполнения DASHBOARD- конфигурации представлен на Рисунок 17.

```
1 import { transferredTable } from "../components/transferredTable";
2 import { actions } from "../components/actions";
3 import { slideMenuFilters } from "../components/sliders";
4
5 const containerStyle = {
6   // необходимые свойства CSS в camelCase
7 }
8
9 export const dashboardTransferredMtrs: any = {
10  displayName: "Сводная информация по переданным в монтаж МТР",
11  pages: {
12    index: {
13      globalStyles: `
14        .PMCBISlideMenuChildren {
15          height: 100%;
16        }
17      `,
18      content: [
19        {
20          type: "container",
21          payload: {
22            style: {
23              // необходимые свойства CSS в camelCase
24            },
25            content: [
26              {
27                type: "container",
28                payload: {
29                  style: {
30                    ...containerStyle,
31                    flexDirection: "column",
32                  },
33                  content: [
34                    actions,
35                    transferredTable,
36                  ],
37                },
38              },
39            ],
40          },
41        },
42        slideMenuFilters,
43      ],
44    },
45  },
46 };
```

Рисунок 17. Пример заполнения DASHBOARD- конфигурация

#### 2.4.6. Описание элементов DASHBOARD- конфигурации

Существуют следующие основные виджеты Composed Chart, Gauge, Kpi, Pie Chart, Table2, Timeline, Tree, компоненты Slide Menu, Filters и другие элементы дашборда (по типу панелей, кнопок и т.п.). Удобнее всего каждый виджет описывать в отдельном файле для декомпозиции кода. А затем из этих файлов импортировать эти элементы в файл с DASHBOARD- конфигурацией и там вставлять их в нужное место описания визуальной структуры.

Пример описания виджета table2 представлен на Рисунок 18.

```
1 import { IDashboardTable } from "@pmc-bi/dashboard-creator";
2 import {
3   fnWhereIdWbs,
4   rowFormatter,
5   onRowClick,
6 } from "../utils/heplers";
7
8 export const transferredTable: IDashboardTable = {
9   name: "transferredTable",
10  type: "table2",
11  rowFormatter: rowFormatter.toString(),
12  onRowClick: onRowClick.toString(),
13  rowGroups: [
14    {
15      sourceTable: "mtr_table",
16      groupBy: "type_supply, mtr_group",
17      where: [
18        {
19          type: "selected",
20          formula: fnWhereIdWbs.toString(),
21        },
22      ],
23      cols: [
24        {
25          formula: "mtr_group",
26          tree: true,
27          displayName: "Группа МТР",
28          alias: "mtr_group",
29        },
30        {
31          formula: "sum(plan_engagement)",
32          displayName: "План вовлечения",
33          alias: "plan_engagement",
34        },
35        {
36          type: "meta",
37          formula: "type_supply",
38          alias: "type_supply",
39        }
40      ],
41    },
42  ],
43 };
```

Рисунок 18. Пример описания виджета table2

#### 2.4.7. Описание основного файла запуска скриптов

В корне папки с наименованием рассматриваемой схемы дашбордов должен располагаться основной файл, который будет запускаться в первую очередь при выполнении скриптов, так называемая точка входа. Это может быть файл index.ts.

Пример такого файла представлен на Рисунок 19.

```
1 import { createDashboard } from "@pmc-bi/dashboard-creator";
2 import { olapConfig, tableConfigs } from "../etl-olap-readmodels/olapConfig";
3 import { etlConfig } from "../etl-olap-readmodels/etlConfig";
4 import { dashboards } from "../dashboards";
5
6 export const main = async () => {
7   await createDashboard(
8     {
9       apiUrl: "https://api.pmc-bi.com",
10      schemaId: "summary-transferred-mtr",
11      schemaDisplayName: "Сводная информация по переданным в монтаж МТР",
12      olapConfig,
13      tableConfigs,
14      etlConfig,
15      dashboardConfigs: dashboards,
16    },
17    {
18      skipConfirm: true,
19      skipETL: true,
20      skipRecreateOlap: true,
21      skipRecreateReadModels: true,
22      skipRecreateDashboard: false,
23    }
24  );
25 };
26
27 main();
```

Рисунок 19. Пример содержания основного файла

В этом файле задается `apiBaseUrl` – адрес бэкенда PMC BI, задается `schemaId` – наименование схемы с дашбордами, импортируются все конфигурации, определенные ранее. А также импортируется и вызывается функция, которая отправляет все эти конфигурации в бэкенд PMC BI.

#### 2.4.8. Запуск скрипта создания дашбордов

В терминале внутри папки `dashboar-creator-xxxx-xx-xx/` запускаем, основной файл, используя `ts-node`, т.к. непосредственно сам Node.js файлы TypeScript запускать не умеет.

Выполняем команду:

```
npx ts-node .\examples\project-name\dashboards-schema-name.ts
```

Например:

```
npx ts-node .\examples\SUIP\summary-transferred-mtr\index.ts
```

Для пересоздания только определенных конфигураций после указания основного файла в этой команде запуска можно указывать флаги через пробел --olap, --etl, --dash, каждый в отдельности или все сразу. Будут пересозданы только указанные конфигурации.

### 2.4.9. Открытие созданных дашбордов

Созданные дашборды будут доступны в браузере по адресу следующего вида:

`http://bi-frontend-ip:bi-frontend-port/#/schema/dashboards-schema-name`

Например:

`http://127.0.0.1:6202/#/schema/summary-transferred-mtr`

Перейдя по такому адресу, на закладке «Дашборды» выбрать интересующий дашборд и нажать кнопку «Открыть».

Пример открывшегося дашборда представлен на Рисунок 20.



Рисунок 20. Пример дашборда

## 2.4.10. Просмотр кубов

Для просмотра кубов после открытия главной страницы необходимо нажать на плитку «Список OLAP-кубов». При этом откроется список всех OLAP-кубов, присутствующих на рассматриваемом сервере.

Кликнув по конкретному кубу, можно открыть список дашбордов, созданных на его основе, а также настройки ETL и прочие настройки.

Пример списка OLAP-кубов представлен на Рисунок 21.

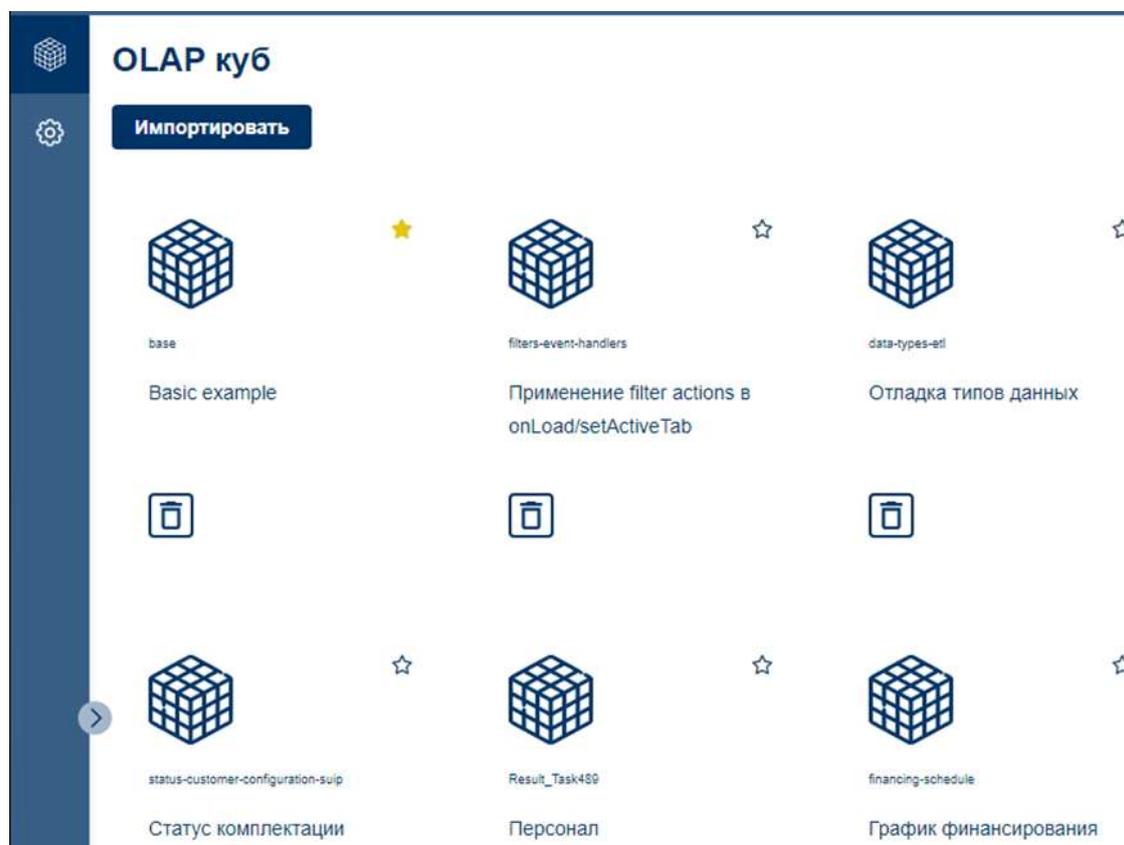


Рисунок 21. Пример списка OLAP-кубов

## 2.4.11. Просмотр настроек ETL

Выбрав конкретный куб и перейдя на вкладку «ETL конфиг», можно увидеть настройки ETL для этого куба.

Пример настроек ETL представлен на Рисунок 22.

Рисунок 22. Пример списка настроек ETL

### 2.4.12. Просмотр процессов ETL

Для просмотра информации о выполненных процессах ETL и состоянии, в котором они находятся, необходимо перейти на вкладку «ETL процессы».

Пример списка процессов ETL представлен на Рисунок 23.

Статус	ID Пайплайна	Старт	Продолжительность	Тип запуска
passed	03304ca3-ad07-42ce-a1fa-a31f93ccc5fc	06.02.2024 13:41:38	1 секунда	manual
passed	38e4f7c1-9240-4b0f-a275-2e3e3c17c8a2	05.02.2024 19:06:27	1 секунда	cron

Рисунок 23. Пример списка процессов ETL

### 2.4.13. Добавление нового отчета

Пример добавления нового отчета представлен на Рисунок 24.

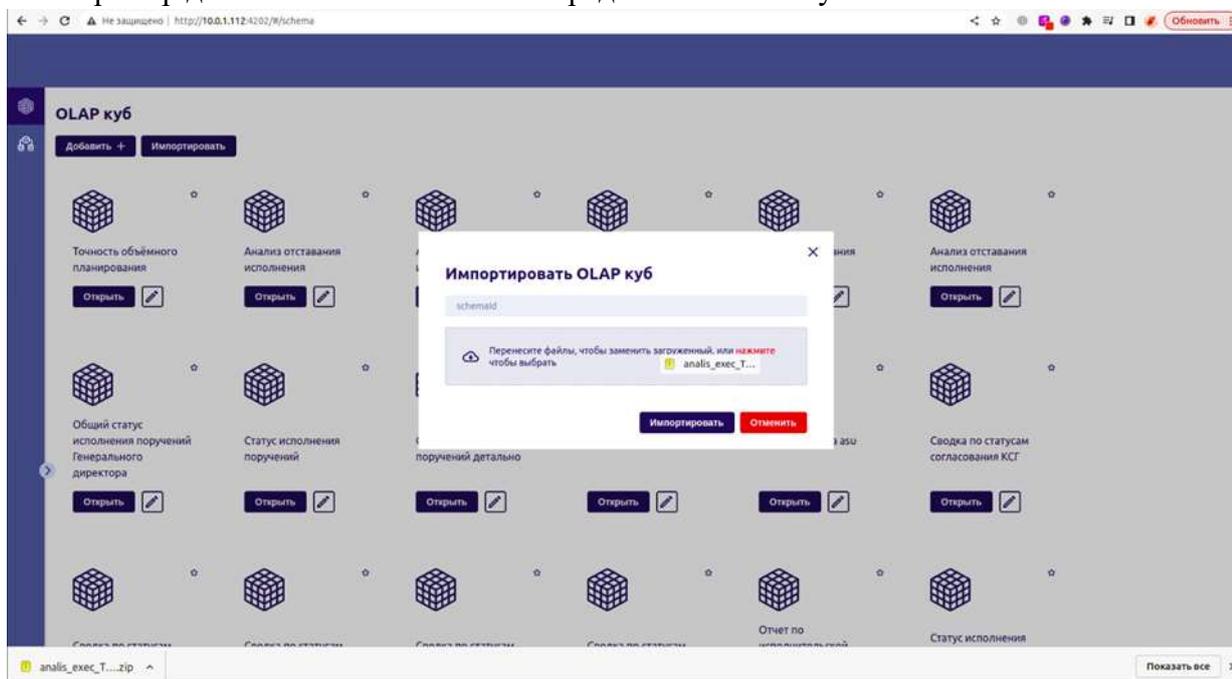


Рисунок 24. Пример добавления нового отчета

### 2.4.14. Экспорт и импорт отчета

Пример экспорта и импорта отчета представлены на Рисунок 25.

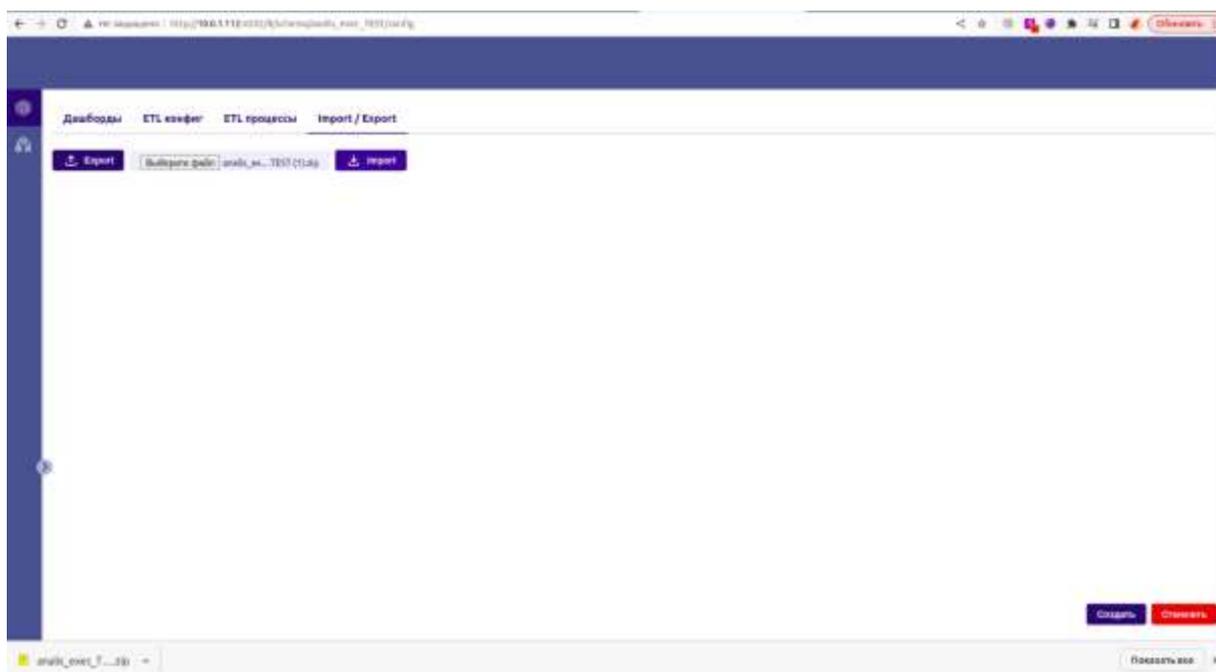


Рисунок 25. Экспорт и импорт отчета