



**ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ ПОСТРОЕНИЯ ХРАНИЛИЩ ДАННЫХ И РАБОТЫ  
С ХРАНИЛИЩАМИ ДАННЫХ И БОЛЬШИМИ ДАННЫМИ (DAT.AX)**

**Инструкция по установке компонент**

**Листов 20**

## СОДЕРЖАНИЕ

СОДЕРЖАНИЕ .....	2
1. Установка компонента DatAx Meta в кластер k8s .....	3
2. Установка компонента DatAx ETL в кластер k8s .....	4
3 Установка через docker-compose компонент DatAx: Meta + LC .....	6
4. Проверка работоспособности компонента DatAx Meta .....	7
4.1. Путь до Dat.ах-интерфейса .....	7
4.2. Настройка профиля пользователя .....	8
4.3. Создание необходимых веток в репозитории metadata.....	8
4.4. Выбор ветки для работы в интерфейсе Dat.ах .....	9
4.5. Загрузка ветки .....	9
4.6. Выбор объекта для редактирования.....	10
4.7. Редактирование объекта.....	11
4.7.1. Добавление строки.....	11
4.7.2. Сохранение внесенных изменений.....	11
4.8. Генерация релиза .....	12
4.9. Создание нового слоя метаданных.....	13
4.10. Просмотр дельты по объекту .....	15
5. Проверка работоспособности компонента DatAx ETL .....	16
5.1. Настройка подключения к БД.....	16
5.2. Создание проекта .....	17
5.3. Создание тестового пайплайна ETL в проекте .....	17
5.4. Создание подключения в Airflow .....	18
5.5. Подтверждение наличия созданного дага в Airflow.....	19

## 1. Установка компонента DataX Meta в кластер k8s

1. Загрузить образы контейнеров в клиентский registry из источника ниже:

а) [Browse - Sonatype Nexus Repository \(ru-central1.internal\)](#)

б) Login: external-puller password: [9:h0EOA6B&b

Список актуальных образов для DataX Meta:

- composite-build:1.0.7-test
- common-admin-panel:1.0.5-test
- common-db-storage:1.0.4-test
- common-file-storage:1.0.4-test
- common-mapping-parser:1.0.4-test
- common-release-generator:1.0.4-test
- common-vcs-storage:1.0.4-test
- connector-db-file:1.0.4-test
- liquibase-object-builder:1.0.4-test
- lc-front:1.0.5-test
- keycloak:latest
- postgres:latest

2. Загрузить приложенный архив с helm чартами.



DataXMeta.zip

Список чартов для приложения DataX Meta:

- composite-build
- common-admin-panel
- common-db-storage
- common-file-storage
- common-mapping-parser
- common-release-generator
- common-vcs-storage
- connector-db-file
- liquibase-object-builder
- lc-front
- keycloak (opensource helm)
- postgres (opensource helm)

3. В каждом чарте актуализировать values.yaml согласно комментариям.

4. По каждому компоненту произвести установку командой «helm install <имя релиза> <путь до чарта>»

5. Создать в кластере k8s:

а) Persistent volume:

- pv-meta-data-logs
- pv-meta-data-srv

б) Persistent volume claim:

- pvc-meta-data-logs
- pvc-meta-data-srv

в) Secret для выкачки образов из собственного registry:

- docker-registry-pull-secret

6. Требуется дополнительно создать:

- Git репозиторий для хранения метаданных
- Git репозиторий для хранения дельты

7. Архив meta\_branches.zip, полученный с поставкой, содержит ветки репозитория метаданных, которые нужно в нем создать. Описание веток представлено в таблице ниже.

Таблица 1 — Описание веток

Репозиторий	Ветка	Источник для первоначального наполнения
datax	meta_values	Распаковать архив meta_values.zip в корневую директорию ветки
datax	meta_templates	Распаковать архив meta_templates.zip в корневую директорию ветки
datax	meta_features	Распаковать архив meta_features.zip в корневую директорию ветки



meta\_branches.zip

## 2. Установка компонента DataX ETL в кластер k8s

1. Загрузить образы контейнеров в клиентский registry из источника ниже:

- а) [Browse - Sonatype Nexus Repository \(ru-central1.internal\)](#)
- б) Login: external-puller password: [9:h0EOA6B&b

Список актуальных образов для чартов:

- lc-back:1.0.5-test
- lc-front:1.0.5-test

- /apache/airflow:2.8.3-python3.10
  - postgres:latest
  - keycloak:latest
2. Загрузить приложенный архив с helm чартами.



DataxETL.zip

Список необходимых чартов для приложения Dat.ах:

- lc-back
- lc-front
- airflow (opensource helm)
- postgres (opensource helm)
- keycloak (opensource helm)

3. В каждом чарте актуализировать values.yaml согласно комментариям.

4. По каждому компоненту произвести установку командой «helm install <имя релиза>

<путь до чарта>»

5. Требуется дополнительно создать:

- а) Git репозиторий для хранения dags airflow.

### 3 Установка через docker-compose компонент DataX: Meta + LC

1. Загрузить образы контейнеров в клиентский registry из источника ниже:

а) [Browse - Sonatype Nexus Repository \(ru-central1.internal\)](#)

б) Login: external-puller password: [9:h0EOA6B&b

2. Создать:

а) Git репозиторий для хранения мета данных

б) Git репозиторий для хранения дельты

в) Git репозиторий для хранения dags airflow.

3. Загрузить приложенный архив с docker-compose:



docker\_compose.zip

4. В папку “./tls/ssl/” положить сертификаты. Прописать их название в tls.yml (“./tls/tls.yml ”)

5. Отредактировать “./docker-compose.yml”

а) В сервисах заменить пути образов в поле `image`

б) Переопределить значения в “./docker-compose.yml” в блоках `enviroment`:

- В сервисах backend, git-sync прописываем актуальный репозиторий и ветку для хранения дагов.

- В сервисах datax-backend, liquibase\_object\_builder в переменной «LOB\_LIQ\_REPO\_URL» прописываем актуальный репозиторий для дельты, а в «LOB\_LIQ\_REPO\_BRANCH\_NAME» ветку. Аналогичный репозиторий указываем в переменной «TC\_BB\_LIQUIBASE\_REPO\_URL», но без данных авторизации.

6. Выполнить запуск приложения с помощью команды `docker-compose up -d`

7. Выполнить вход на UI (**Ошибка! Недопустимый объект гиперссылки.**) login: “test\_user” password: “1234”, проверить работоспособность приложения.

Docker-compose содержит следующие сервисы:

- airflow-init
- airflow-scheduler
- airflow-triggerer
- airflow-websserver
- airflow-worker
- git-sync
- keycloak

- redis
- reverse-proxy
- postgres\_db
- backend
- frontend
- common\_admin\_panel
- common\_db\_storage
- common\_file\_storage
- common\_mapping\_parser
- common\_release\_generator
- common\_vcs\_storage
- connector\_db\_file
- liquibase\_object\_builder
- datax-backend

#### 4. Проверка работоспособности компонента DatAx Meta

Важно при первичном запуске активировать русский язык в профиле.

##### 4.1. Путь до Dat.ах-интерфейса

Для того, чтобы зайти в раздел Dat.ах, необходимо в верхней левой части интерфейса открыть меню (Рис. 1) и выбрать раздел «Управление метаданными».

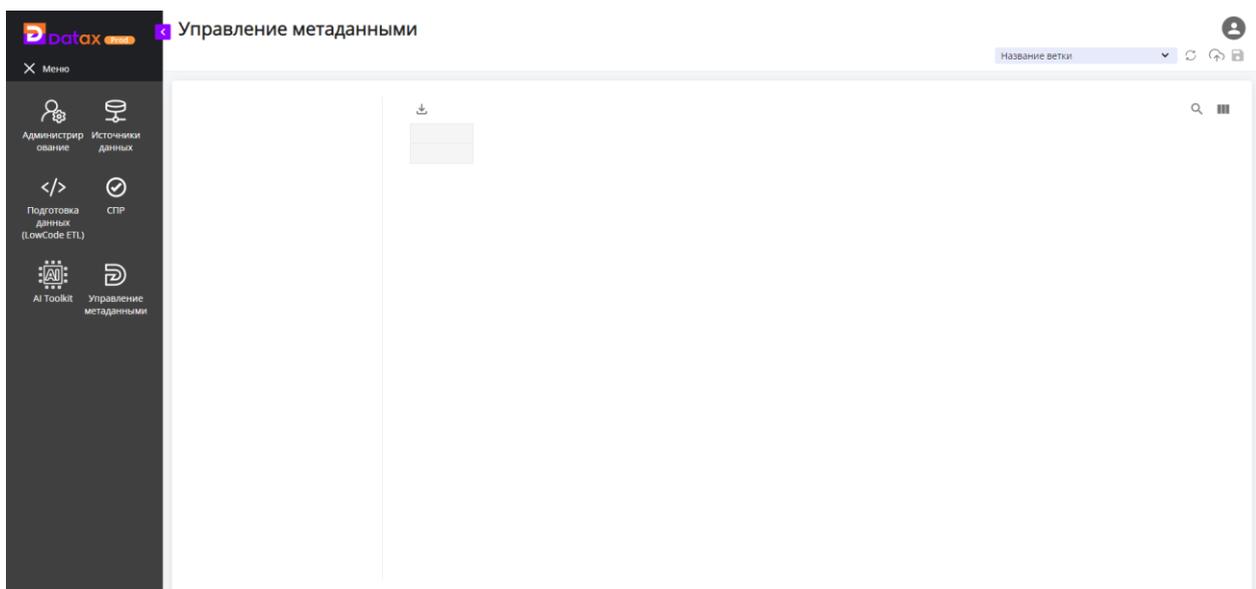


Рис. 1 Главное меню

## 4.2. Настройка профиля пользователя

В верхнем правом углу иконка пользователя . После нажатия открывается окно профиля пользователя.

Чтобы настроить профиль пользователя необходимо нажать на иконку  и заполнить данные (Рис. 2):

- Логин — логин, использующийся для GitLab;
- Токен – токен, созданный в GitLab;
- Репозиторий – добавить ссылку на репозиторий (2).

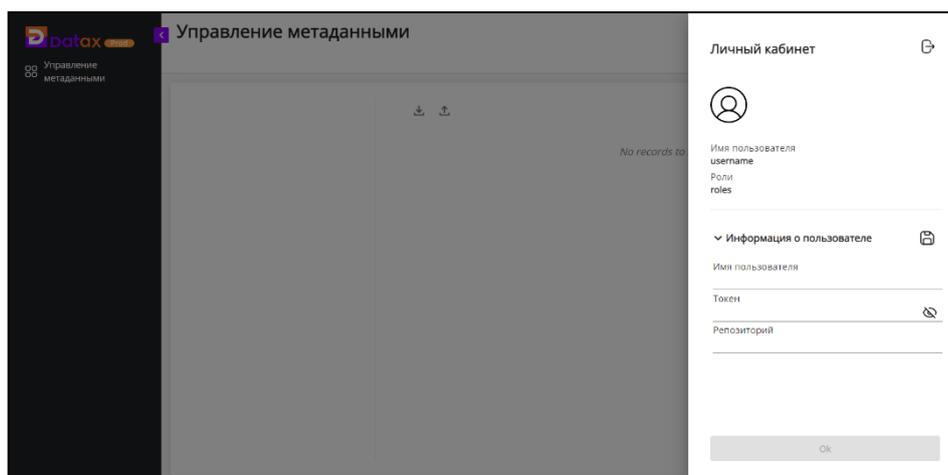


Рис. 2 Профиль пользователя

После настройки необходимо еще раз нажать на иконку  для сохранения изменений и нажать «ОК».

## 4.3. Создание необходимых веток в репозитории metadata

Архив meta\_branches.zip, полученный с поставкой, содержит ветки репозитория метаданных, которые нужно в нем создать. Описание веток представлено в таблице ниже.

Репозиторий	Ветка	Источник для первоначального наполнения
datax	meta_values	Распаковать архив meta_values.zip в корневую директорию ветки
datax	meta_templates	Распаковать архив meta_templates.zip в корневую директорию ветки
datax	meta_features	Распаковать архив datax_docker.zip в корневую директорию ветки



meta\_branches.zip

#### 4.4. Выбор ветки для работы в интерфейсе Dat.ah

Выбор ветки осуществляется в верхней правой части интерфейса (Рис. 1).

Необходимо нажать на «название ветки» .

Появится выпадающий список, из которого можно выбрать ветку meta\_values (Рис. 3).

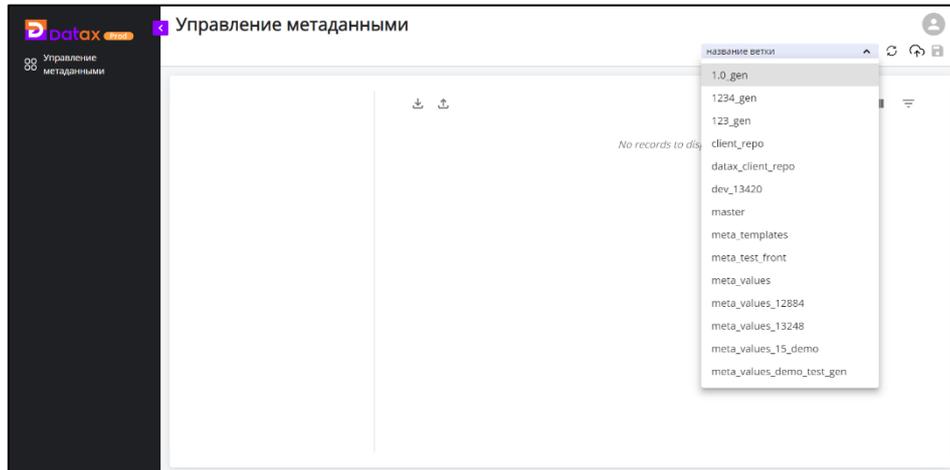


Рис. 3 Выбор ветки

#### 4.5. Загрузка ветки

После появления выпадающего списка выберите ветку и нажмите на нее.

Выбранная ветка загружается в систему (Рис. 4).

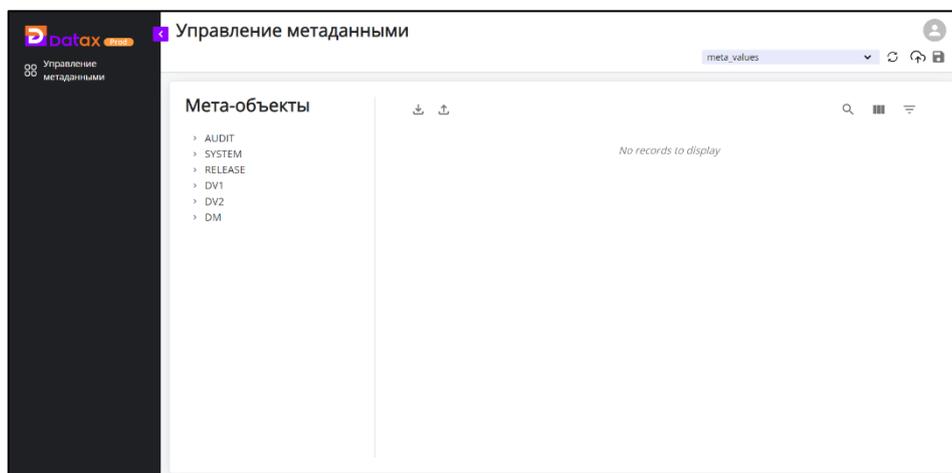


Рис. 4 Загрузка ветки

Ветка успешно загрузилась, в интерфейсе появилась таблица «Мета-объекты» со списком доступных сущностей (Рис. 5).

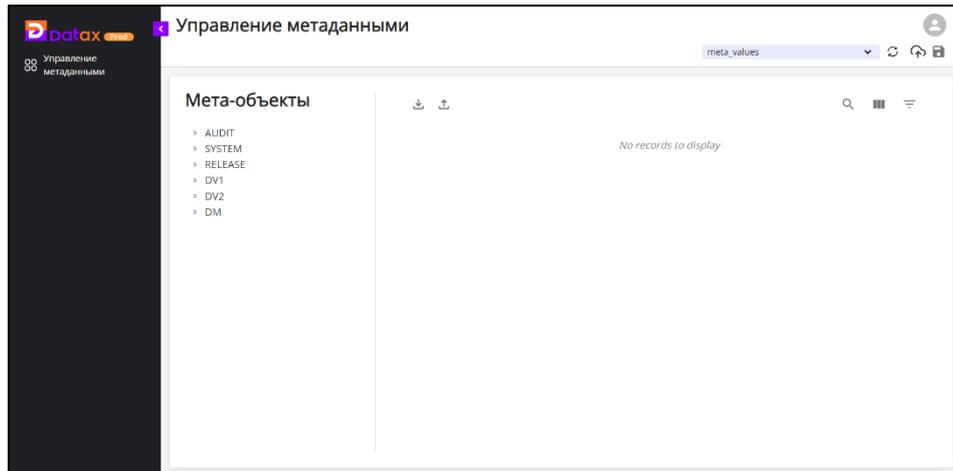


Рис. 5 Мета-объекты

#### 4.6. Выбор объекта для редактирования

После загрузки ветки слева под надписью «Мета-объекты» появился список (Рис. 5).

В списке можно выбрать требуемую модель. При нажатии на нужную модель появится список сущностей модели (Рис. 6).

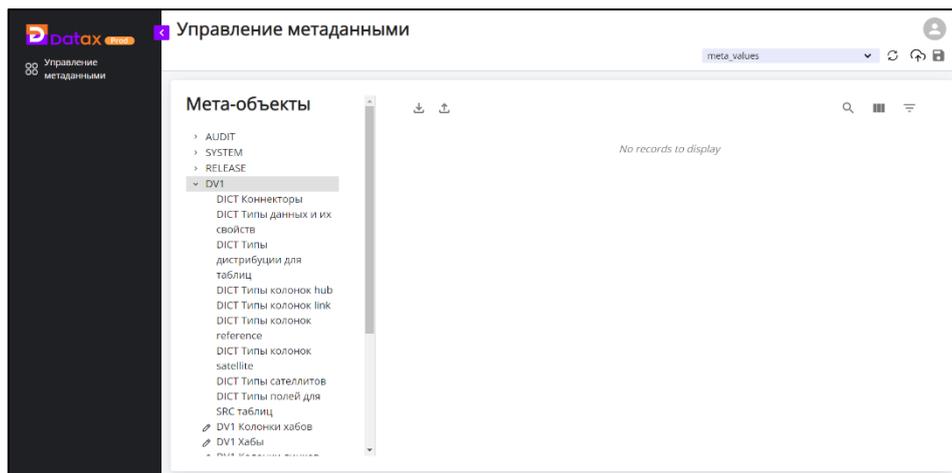


Рис. 6 Список сущностей модели

Сущности типа DICT невозможно изменять. Сущности, в которые пользователь может вносить изменения помечены значком .

После выбора в списке нужной сущности, щелкнув на нее, справа появятся внутреннее содержание таблицы (Рис. 7).

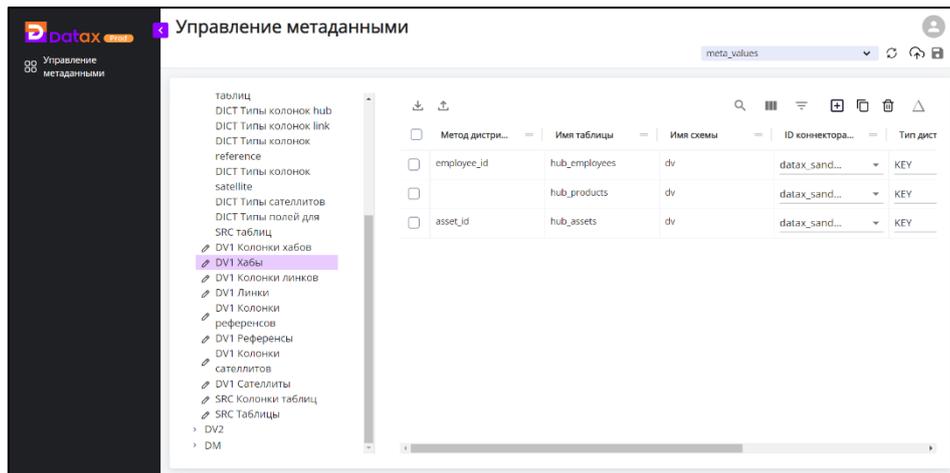


Рис. 7 Внутреннее содержание сущности

## 4.7. Редактирование объекта

### 4.7.1. Добавление строки

Чтобы добавить строку, необходимо нажать кнопку «Добавление» (Рис. 8).

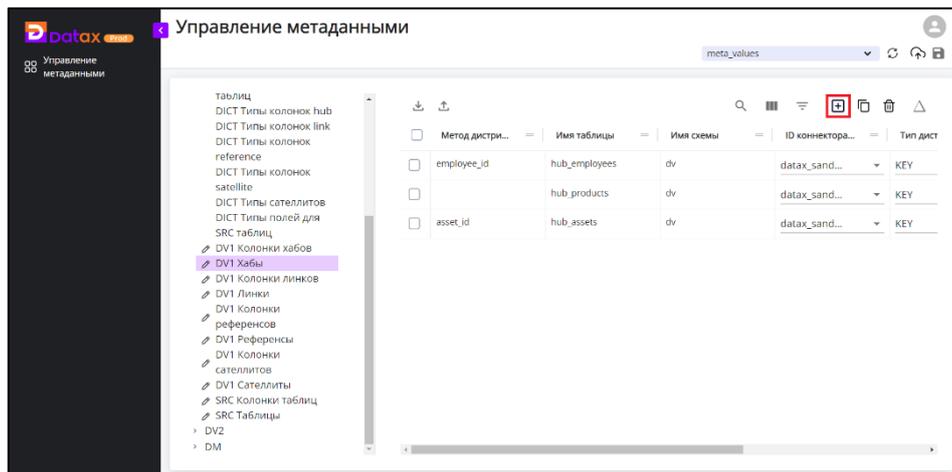


Рис. 8 Добавление строки

### 4.7.2. Сохранение внесенных изменений

Для сохранения внесенных изменений необходимо нажать на появившуюся кнопку «Сохранить» (Рис. 9).

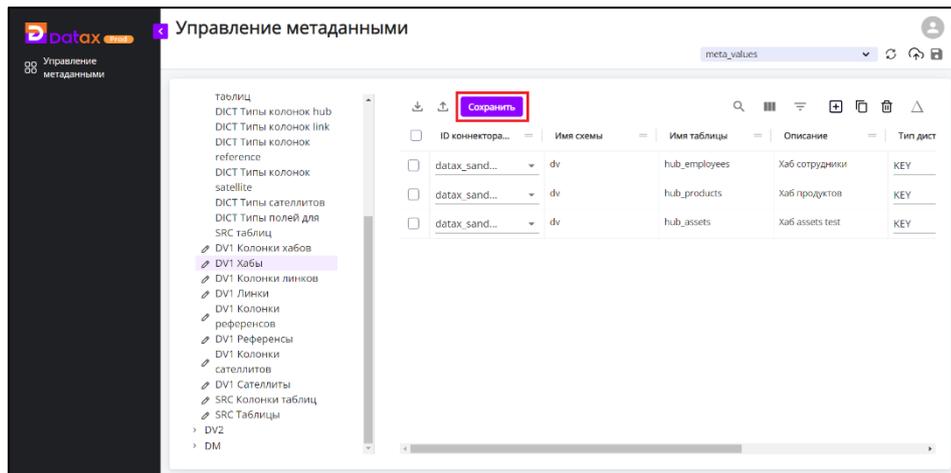


Рис. 9 Сохранение данных

#### 4.8. Генерация релиза

Для генерации релиза нужно нажать на иконку «Сохранить ветку»  и выбрать пункт «Сгенерировать артефакты» (Рис. 10).

Генерация артефактов

Commit: [Task ID][Краткое описание изменений]

Информация о пользователе

Area

ID Задачи

ID Релиза

Сгенерировать s2t файлы

Дельта

Отменить **COMMIT**

Рис. 10 Генерация релиза

В открывшемся окне заполнить необходимые данные, можно произвольными значениями:

- Commit - [Task ID][Краткое описание изменений]
- Информация о пользователе
- Area
- ID Задачи
- ID Релиза

Затем необходимо нажать кнопку Commit. Получаем сообщение об успешной генерации артефактов (Рис. 11).

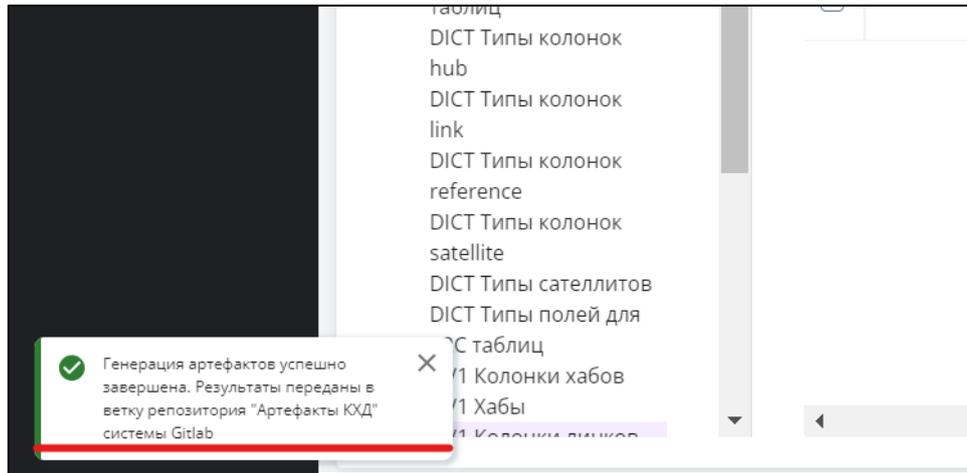


Рис. 11 Сообщение об успешной генерации релиза

#### 4.9. Создание нового слоя метаданных

Перейдите в раздел «Администрирования», выберите пункт «Настройка моделей метаданных» (Рис. 12).

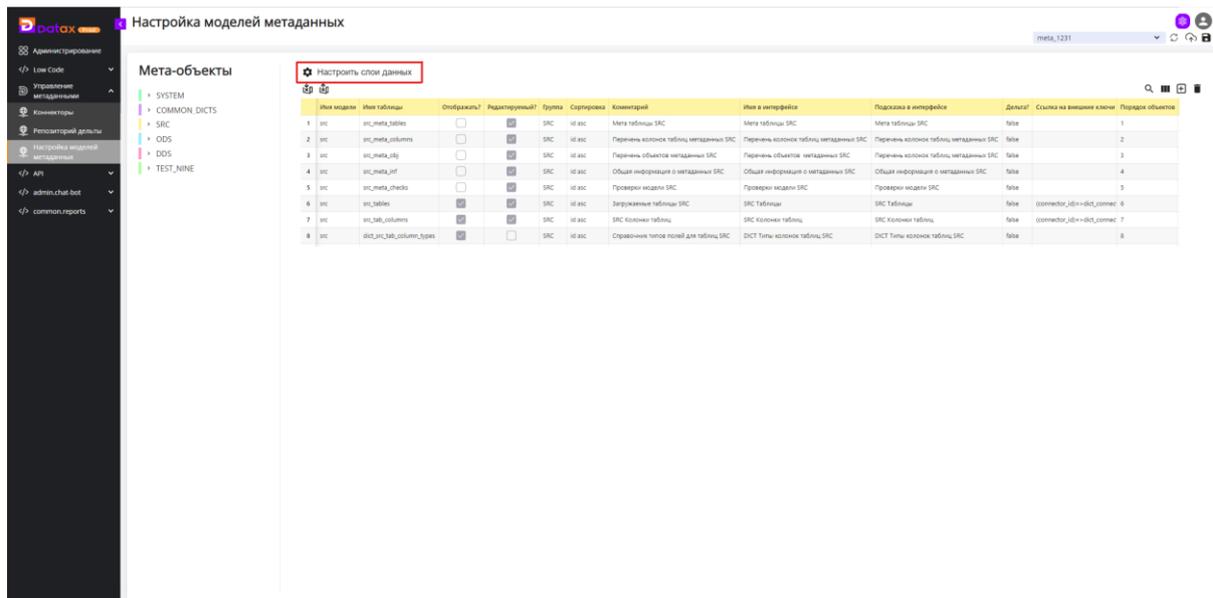


Рис. 12 Настройка моделей метаданных

Нажмите на кнопку «Настроить слои данных», откроется окно «Добавление модели метаданных» (Рис. 13).

Добавление модели метаданных

название слоя

Название шаблона

Отменить

ДОБАВИТЬ

Рис. 13 Добавление модели метаданных

Укажите название слоя и выберите один из предложенных шаблонов (Рис. 14).

Добавление модели метаданных

название слоя  
TEST\_NINE

Название шаблона  
Third Normal Form

Отменить

ДОБАВИТЬ

Рис. 14 Добавление модели метаданных, заполненное окно

После нажмите кнопку «Добавить». Спустя некоторое время будет создан новый слой, который можно будет настраивать.

Если приложение DataX Meta поставляется без функционала расчета дельты по объектам, то проверку 1.10 можно пропустить.

#### 4.10. Просмотр дельты по объекту

Чтобы просмотреть дельту по объекту, необходимо открыть таблицу, по которой хочется просмотреть дельту. Выбрать строку, по которой нужно просмотреть дельту, иконка  будет доступна. Кликнув на иконку, появится выбор:

- Дельта (Рис. 15)

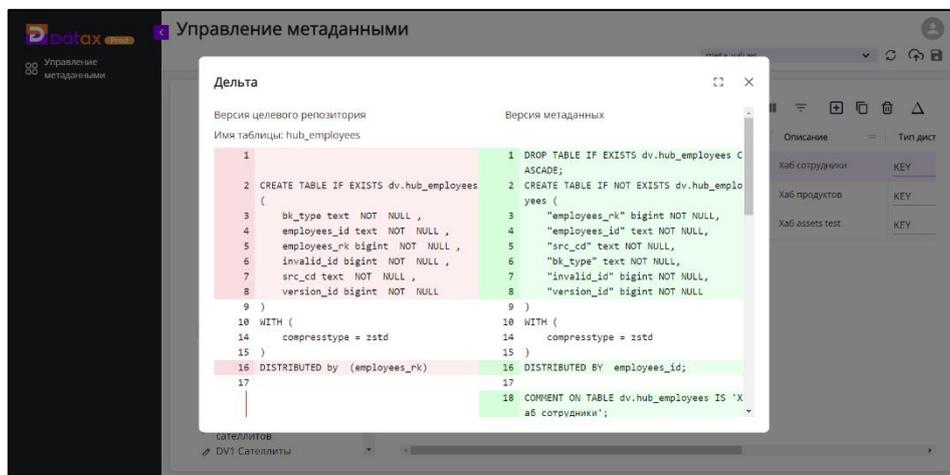


Рис. 15 Дельта

- Дельта с источниками (Рис. 16)

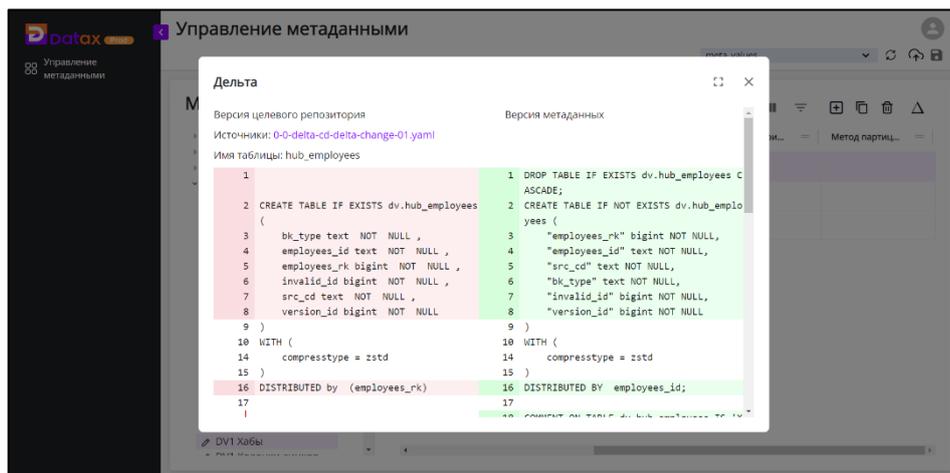


Рис. 16 Дельта с источниками

Отличие в том, что «Дельта с источниками» предоставляет ссылку на файл целевого репозитория, где описана выбранная сущность.

Чтобы развернуть результат на весь экран, нужно нажать на иконку  (Рис. 17).

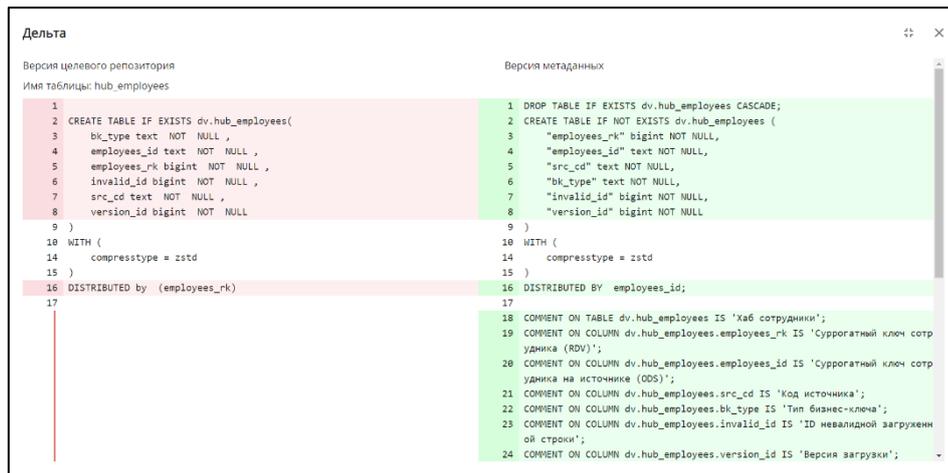


Рис. 17 Дельта на весь экран

## 5. Проверка работоспособности компонента DataX ETL

### 5.1. Настройка подключения к БД

Для создания нового подключения перейдите во вкладку «Источники данных» (Рис. 18)

и нажмите кнопку 



Рис. 18 Добавление нового подключения к БД

Укажите все необходимые данные в форме «Создание подключения» (Рис. 19) и активируйте переключатель `DBMSConnection.default_stage_connection`, нажмите кнопку «Сохранить».

Создание подключения ☰ ×

Тип БД  
Postgres

Наименование

Описание

Схема  
public

Host

Название БД

Port  
80

Имя пользователя

Пароль

ID подключения в AIRFLOW

DBMSConnection.default\_stage\_connection

Использовать для загрузки xlsx

[Проверить](#) [Отменить](#) [Сохранить](#)

Рис. 19 Создание подключения

Убедитесь в корректном подключении, нажав кнопку «Проверить».

## 5.2. Создание проекта

Заходим на вкладку «Подготовка данных» и нажимаем кнопку . Задаем имя проекта и его описание. Нажимаем кнопку «Сохранить».

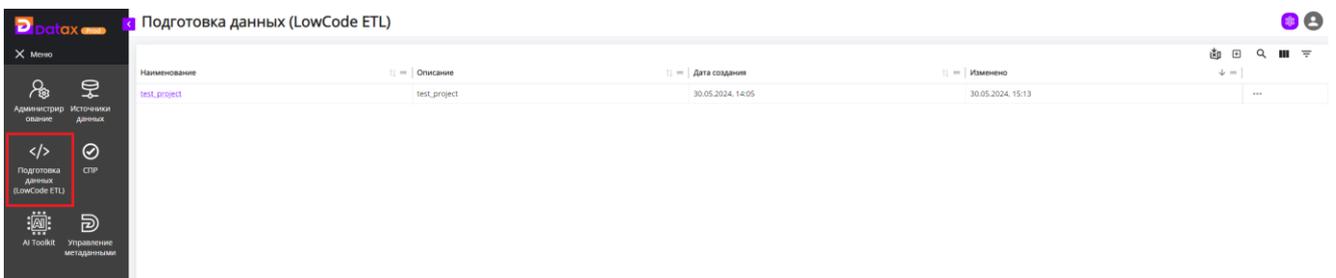


Рис. 20 Интерфейс окна «Подготовка данных»

## 5.3. Создание тестового пайплайна ETL в проекте

Для создания пайплайна переходим в проект, созданный на предыдущем шаге. В блоке «Источники данных» выбираем модуль «SQL» (зеленого цвета) и перетягиваем его на интерактивное поле.

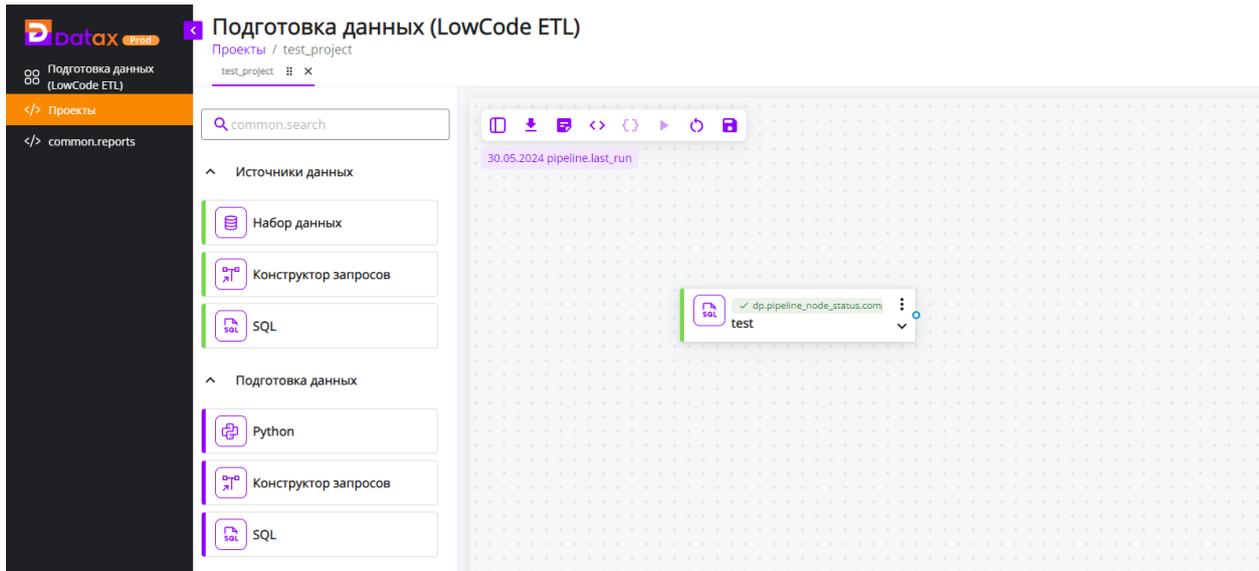


Рис. 21 Создание простого пайплайна

Переходим в меню настройки модуля (Рис. 22).

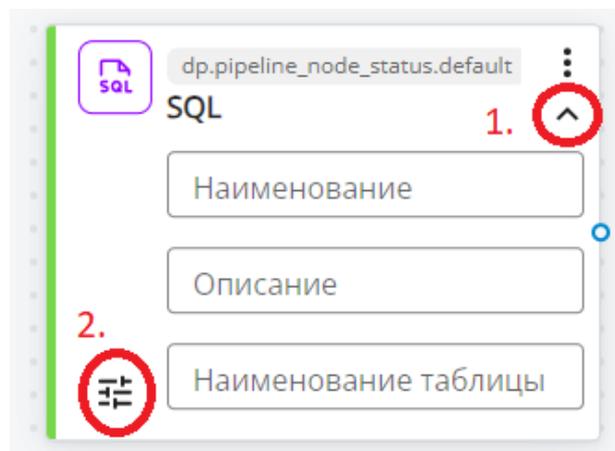


Рис. 22 Последовательность действий для перехода в меню настройки модуля

В выпадающем списке «Подключение» выбираем созданное в пункте 5.2 подключение к БД. В поле запроса прописываем «SELECT '2'» и нажимаем кнопку проверить. Нажимаем кнопку «Сохранить».

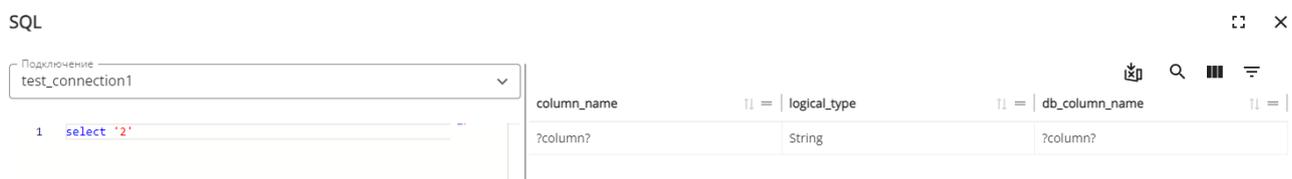


Рис. 23 Настройка модуля

#### 5.4. Создание подключения в Airflow

В верхней панели наводимся на вкладку Admin и переходим в раздел «Connections»  
Создаем подключение, нажав кнопку .

Conn Id	Conn Type	Description	Host	Port	Is Encrypted	Is Extra Encrypted
ID_AIRFLOW	postgres		10.131.1.7	7777	False	False

Рис. 24 Вкладка Connection в Airflow

Описание параметров:

- Connection Id — указываем значение из «ID подключения к AIRFLOW»
- Connection Type — выбираем Postgres
- Host — IP\FQDN хоста на котором развернута БД
- Database — airflow
- Login — airflow
- Password — airflow
- Port — 7777
- Extra — оставляем пустым. **Важно удалить «{}»**

Рис. 25 Вкладка Connection в Airflow, окно Add Connection

### 5.5. Подтверждение наличия созданного дага в Airflow

Переходим в Airflow по адресу <http://<hostname>:8081>, в качестве логина и пароля указываем “airflow”. Переходим на вкладку DAGs. При успешном выполнении предыдущих пунктов в окне отобразится созданный даг.

The screenshot displays the Airflow web interface for the DAGs section. At the top, there is a navigation bar with the Airflow logo and menu items: DAGs, Cluster Activity, Datasets, Security, Browse, Admin, and Docs. The current time is 12:17 UTC. Below the navigation bar, the 'DAGs' section is active, showing a filter for 'test\_project' and a search bar. The main content is a table with the following columns: DAG, Owner, Runs, Schedule, Last Run, Next Run, Recent Tasks, Actions, and Links. Two DAGs are listed:

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actions	Links
NODE_DAG_584d5a10-383c-4693-89cd-d0c424f6fb23 <a href="#">test_project</a>	test_user	0	None			0	[Stop] [Refresh] [More]	
PIPELINE_DAG_bc5d4451-1c79-4839-a334-4431f66d620d <a href="#">test_project</a>	test_user	1	None	2024-05-30, 15:13:18		1	[Stop] [Refresh] [More]	

At the bottom right, it says 'Showing 1-2 of 2 DAGs'.

Рис. 26 Вкладка DAGs в Airflow