Программное обеспечение

Система управления данными Polyflow

РУКОВОДСТВО РАЗРАБОТЧИКА

Аннотация

Настоящий документ является руководством разработчика системы управления данными Polyflow.

Документ разработан в соответствии с требованиями ГОСТ Р 59795-2021 «Требования к содержанию документов».

Содержание

Введение

Модуль Polyflow представляет собой сервис оркестровки, сбора и обработки разнородных данных в хранилищах произвольной архитектуры.

Уровень подготовки персонала, необходимого для работы с ИАС, предполагает наличие следующих групп пользователей:

- Служба эксплуатации ИАС;
- Разработчики;
- Операторы.
- Служба эксплуатации ИАС

В службу эксплуатации ИАС входят специалисты следующих категорий: «Администратор защиты (безопасности) информации», «Администратор операционных систем», «Администратор баз данных».

- Администратор защиты (безопасности) информации обеспечивает:
    - Формирование списка пользователей, допущенных к работе с Системой;
    - Настройку учетных записей пользователей, управление ролями доступа, а также интеграция пользователей с помощью LDAP;
    - Формирование матрицы доступа к ресурсам Системы и данным, а также изменение прав доступа.
- Администратор операционных систем отвечает за:
    - Установку компонентов платформы, активацию и первоначальную настройку.
    - Сопровождение ИАС (тестирование работоспособности, восстановление и т.п.), обновление версий (анализ необходимости перехода на новые версии, разработку перечня мероприятий по переводу на новую версию).
- Администратор баз данных отвечает за:

- Генерацию систем управления базами данных;
- Сопровождение и управление информационными ресурсами;
- Сохранение резервных копий, восстановление искаженной информации, архивирование информации и организацию поступления информации из архива;
- Обработку и анализ статистической информации о характере и интенсивности использования данных, о распределении нагрузки на различные компоненты структуры баз данных, внесение изменений в структуру баз данных в процессе эксплуатации Системы с целью повышения производительности, обеспечивает ввод и поддержание в актуальном состоянии общих разделов баз данных (классификаторов).

Служба эксплуатации обеспечивает функционирование в штатном режиме технических и программных средств АИС, отслеживает процессы наполнения АИС данными.

Поддержка функционирования Системы должна осуществляться силами действующей Службы эксплуатации АИС, состоящей из специалистов, обладающих знаниями в области информационных и сетевых платформ, на которых реализована АИС, и опытом администрирования баз данных.

Разработчики

Разработчиками являются специалисты, которые участвуют в процессах разработки процессов загрузки данных в Систему, описания сущностей-получателей и источников, проведения анализа работы процессов, выявления аномалий и их причин.

Разработчики должны иметь опыт разработки в своей отрасли, обладать навыками работы с DWH решениями и базовыми знаниями SQL и Python.

Операторы

Операторами являются специалисты, которые участвуют в процессах сбора данных, базовой настройки, запуска и мониторинга таких процессов.

Для работы с системой необходимо ознакомиться со следующим набором эксплуатационной документации:

- Руководство администратора;
- Руководство разработчика;
- Руководство пользователя.

1 Назначение и условия применения

1.1 Назначение системы

Модуль Polyflow предназначен для решения следующих задач:

- сбор данных из разных источников:
  - файлы
  - внешние системы
  - базы данных
- описание сущностей-получателей и источников;
- мониторинг и управление выполняемыми процессами;
- контроль качества данных;
- трансформация данных.

1.2 Условия применения

Для функционирования АИС необходимо следующее программно-аппаратное обеспечение:

1.2.1 Серверная часть

Минимальные требования к серверному оборудованию следующие:

- CPU 2 vCPU (2.8 ГГц и выше);
- RAM 8 ГБ,
- HDD 10 ГБ

Ориентировочная формула для подсчета конфигурации в зависимости от количества процессов при использовании Local Executor: дополнительно к минимальным системным требованиям необходимо RAM 256-512МБ CPU 0.1 vCPU в среднем на каждый процесс.

Операционная система: Astra Linux Special Edition 1.6 (Смоленск) или аналог.

Права пользователя, разворачивающего приложение: user - non-root with sudo privileges.

Дополнительные требования к установленным приложениям: Docker версии 20.10.0 и до 25, Docker -compose версия 1.29 и выше.

1.2.2 Локальная сеть

Все компоненты платформы должны находиться в одной подсети или должна обеспечиваться прозрачная маршрутизация. Не рекомендуется использовать NAT. В рамках ознакомления рекомендуется отключить брандмауэры. Внутри локальной сети между всеми компонентами не должно быть ограничений по передаче данных. Для доступа из внешней сети достаточно открыть порт, используемый Polyflow (порт задается при установке). При использовании системы с установленными антивирусами или комплексными системами защиты необходимо обеспечить свободную работу, сетевую активность и взаимодействие компонентов.

2 Подготовка к работе

2.1 Состав программного обеспечения

Модуль Polyflow поставляется в виде нескольких файлов:

- образы Docker, содержащие в себе все компоненты с уже настроенным окружением и всеми внутренними зависимостями;
- python-файл manage.py для быстрого развёртывания и настройки модуля;
- дополнительные файлы, используемые при развертывании: __init__.py, docker-compose.yml.tmpl, airflow.cfg.tmpl.

Примечание: Docker — программное обеспечение для автоматизации развёртывания и управления приложениями в среде виртуализации на уровне операционной системы. Суть и одно из предназначений Docker такое же, как и у виртуальных машин — это изоляция работы различных конфликтующих программ внутри одного сервера. Наглядно увидеть отличия между виртуальной машиной и контейнером можно, пройдя по ссылке: https://www.docker.com/what-docker.

Таким образом, установка сводится к двум шагам:

1. Установка docker-engine внутри операционной системы;

2. Запуск python-файла manage.py для разворачивания готовых контейнеров с компонентами системы.

Установка и настройка системы описаны в документе «Руководство администратора». Основные операции с интерфейсом описаны в документе «Руководство пользователя».

2.2 Запуск системы

2.2.1 Начало работы

Для того, чтобы начать работу в Polyflow необходимо произвести аутентификацию в системе. Для этого введите в браузере адрес машины, на котором установлено решение, например https://127.0.0.1:8080 (порт

указывается тот, который был задан при установке). На появившейся форме укажите свой логин и пароль (пароль администратора задаётся при установке системы, пароль пользователя – получается у администратора), и нажмите кнопку «Вход» в соответствии с Рисунок 1.

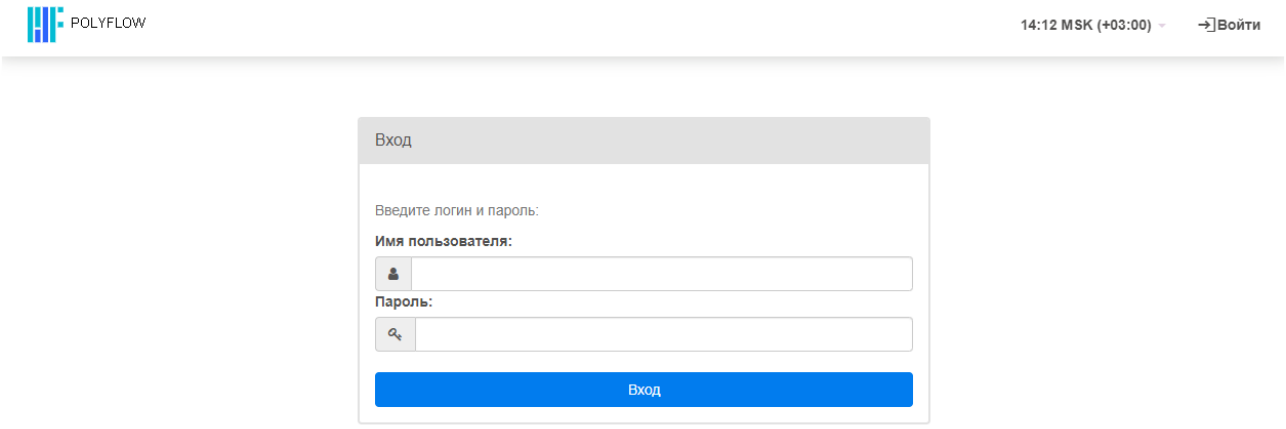

Рисунок 1. Начальная страница

2.3 Порядок проверки работоспособности

Для проверки работы Polyflow необходимо произвести аутентификацию в системе. После успешной аутентификации откроется главная страница приложения, со списком доступных процессов (Рисунок 2). Необходимо выбрать один из них и перейти на форму просмотра информации о выбранном процессе, кликнув на его название (Рисунок 3).

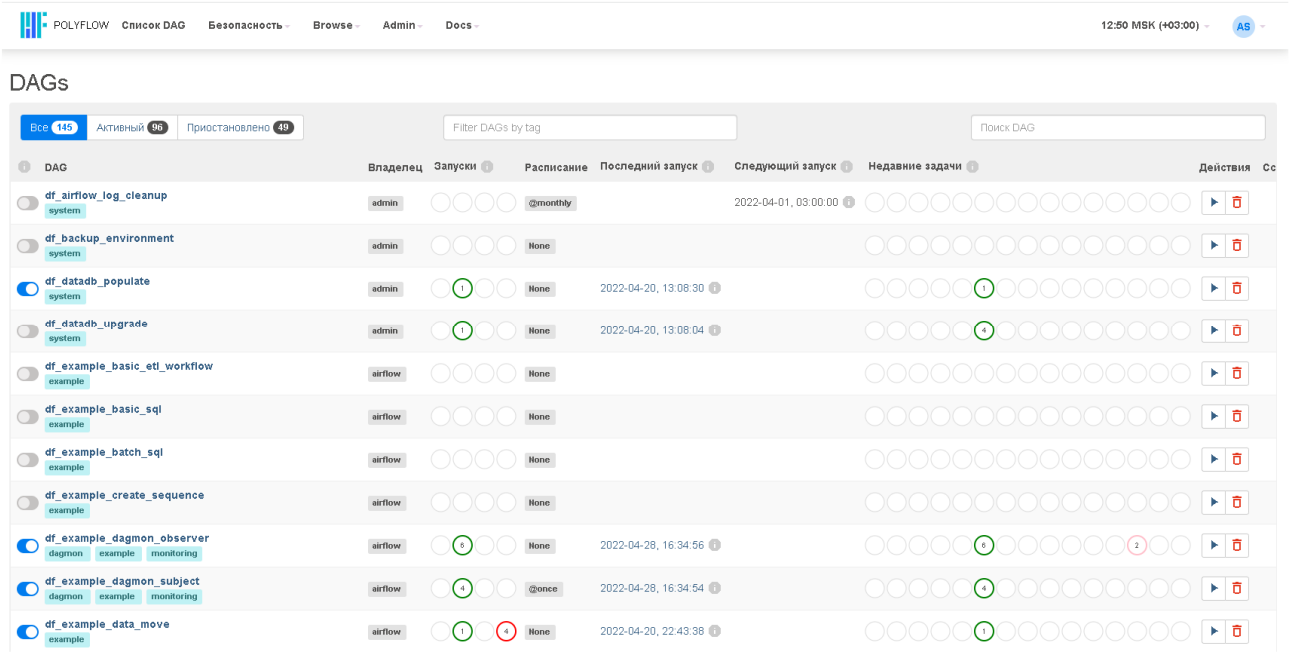

Рисунок 2. Список доступных процессов

Будет открыта форма информации и управления выбранным DAG. По умолчанию открыта вкладка График (Рисунок 3).

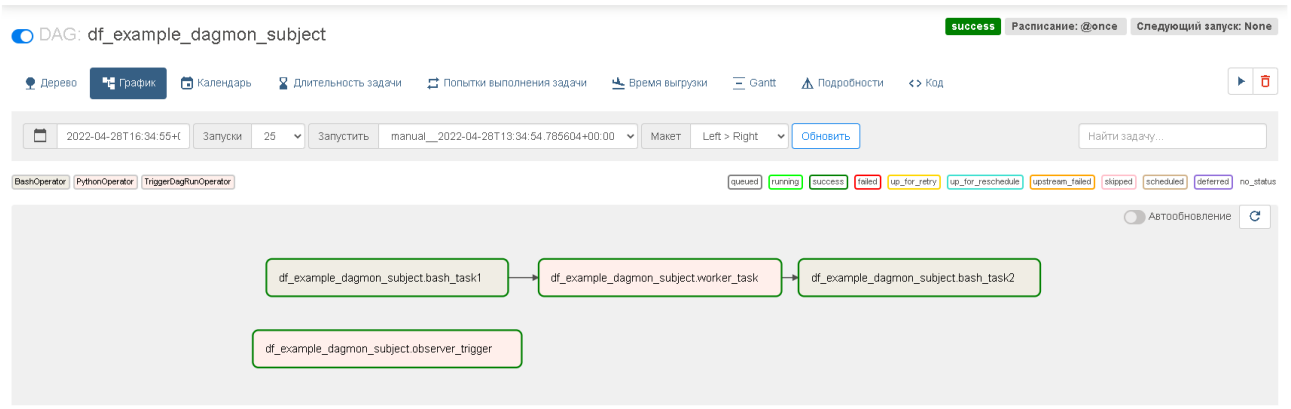

Рисунок 3. Форма просмотра информации о процессе

3 Описание операций

3.1 Определения и сокращения Polyflow

Определения и сокращения Polyflow представлены в таблице 1.

Таблица 1. Определения Polyflow

Термин/Сокращение	Определение
Аутентификация	Проверка принадлежности пользователю указанного им пароля.
Пользователь	Авторизованный пользователь, учетная запись которого позволяет просматривать данные на портале.
Веб-интерфейс	Сайт в компьютерной сети, который предоставляет пользователю интерактивный интернет-сервис, который работает в рамках этого сайта.
Планировщик (Airflow Scheduler)	Компонент, который отслеживает состояние DAG и запускает задачи, зависимости которых были удовлетворены. После запуска системы планировщик работает непрерывно, чтобы отслеживать и синхронизировать папку, содержащую объекты DAG.
Хранилище данных (англ. Content Repository, Data Warehouse, DWH)	Предметно-ориентированная информационная база данных, сочетающая в себе функции системы управления версиями, поисковой машины и СУБД.
Система управления данными Polyflow	Сервис оркестровки сбора и обработки разнородных данных хранилища произвольной архитектуры.
Polyflow	Краткое наименование программного обеспечения «Система управления данными Polyflow»
DAG (Directed Acyclic Graph)	Смысловое объединение задач, которые необходимо выполнить в строго определенной последовательности согласно указанному расписанию.
Task	Операции, применяемые к данным, например: загрузка данных из различных источников, их

	агрегирование, индексирование, очистка от дубликатов, сохранение полученных результатов и прочие ETL-процессы.
OpenID	Открытый стандарт децентрализованной системы аутентификации, предоставляющей пользователю возможность создать единую учётную запись для аутентификации на множестве не связанных друг с другом интернет-ресурсов, используя услуги третьих лиц.
Operator	Сущность, на основе которой создаются экземпляры заданий, где описывается, что будет происходить во время исполнения экземпляра задания.
Sensor	Тип оператора, позволяющий описывать реакцию на определенное событие.
ETL	(от англ. Extract, Transform, Load – дословно «извлечение, преобразование, загрузка») – один из основных процессов в управлении хранилищами данных.
SLA	(от англ. Service Level Agreement) – соглашение об уровне сервиса.

4 Принципы реализации процессов на Polyflow

4.1 Общие

Все проектные артефакты должны храниться в репозитории, локальном либо централизованном. Комментарии в коде и к коммитам должны быть на русском языке. В README к проекту должны быть указана основная ветка репозитория, если это не master (main). Секреты стендов в промышленной эксплуатации в репозиторий коммититься не должны. Для подключения к внешним системам должны использоваться Airflow Connection, а не переменные окружения, константы или иные, небезопасные plain-text источники. При наличии системных модулей всегда использовать их через импорт, а не копированием. Доработки, необходимые более чем для одного проекта рекомендуется реализовывать как системные. Типовые процессы, которые можно использовать как шаблоны при реализации собственных, представлены в `volumes/workspace/dags/df/examples` и `volumes/workspace/dags/df/tests`.

4.1.1 ETL

Все задачи по обработке данных должны выполняться в `DockerOperator` (образ - `df_operator`). Остальные операторы и сенсоры допустимо использовать для задач следующих типов: проверка существования ресурсов (`LocalFileSensor`), перемещение/удаление файловых объектов (`BashOperator`), контроль задач (`TriggerDagRunOperator`, `ExternalTaskSensor`, `LatestOnlyOperator`), операции управления планами загрузки Visiology (`VisiologyAPIOperator`), нетребовательные вспомогательные операции (`PythonOperator` и производные от него операторы управления workflow). Поддержка прочих операторов не осуществляется. Не допускается установка дополнительных модулей в образы `df_airflow`, ни в базовый, ни в локальный, как действие, приводящее к необходимости внеплановых обновлений и перезапусков сервиса, конфликтам модулей,

падениям, зависаниям, внутрисистемной конкуренции процессов и общему снижению стабильности работы системы.

4.1.2 DWH

Все сущности базы данных должны быть описаны метаданными и созданы на их основе средствами Polyflow. Создание сущностей скриптами допускается только в случае невозможности их описания в метаданных. На отсутствующий функционал должна быть запрошена доработка, создана соответствующая задача в Polyflow. Изменения в структуре сущностей должны быть отражены в метаданных, сами изменения при этом могут, а в некоторых случаях и должны, выполняться руками.

4.2 Описание файлов и структуры проекта

Вся работа происходит в папке projects, содержимое которой линкуется с папкой workspace.

Для инициализации структуры проекта необходимо выполнить команду python3 manage.py --init-project <Имя проекта>. Если имя проекта не задано, то будет использовано значение default. После выполнения команды будут созданы следующие папки:

- `dataflow/volumes/projects/<Имя проекта>/cache` - директория для загруженных/закэшированных файлов
- `dataflow/volumes/projects/<Имя проекта>/dags` - директория для дагов
- `dataflow/volumes/projects/<Имя проекта>/metadata` - директория для метаданных сущностей и прочих объектов
- `dataflow/volumes/projects/<Имя проекта>/plugins` - директория для плагинов
- `dataflow/volumes/projects/<Имя проекта>/share` - директория для данных для обработки

- `dataflow/volumes/projects/<Имя проекта>/source` - директория для исходников

4.2.1 Структура исходников

- `source` - основная директория
  - `df` - директория с системным кодом проекта (библиотеками), внутри которой располагаются следующие:
    - `common` - директория с общим кодом проекта (библиотеками)
    - `excel` - экстракт из датасетов в табличном формате (excel, netdb)
    - `python` – различные раннеры
    - `qc` - директория объектов контроля качества
    - `sql` - директория sql объектов:
      - `scripts` – выполняемые скрипты
      - `ddl` – скрипты миграции
      - `callables` – процедуры

- `metadata` - основная директория метаданных, может представлять собой иерархию
- `dags` - основная директория, может представлять собой иерархию, но в веб-интерфейсе всегда имеет плоский вид, где видимость DAG'ов регулируется правами
  - `examples` - директория с примерами настройки DAG'ов, можно скрыть из веб-интерфейса с помощью файла `.airflowignore` в директории `volumes`
  - `tests` - директория с тестовыми DAG'ами, можно скрыть из веб-интерфейса с помощью файла `.airflowignore` в директории `volumes/workspace/dags`

Для сохранения своей версии `.airflowignore` можно воспользоваться следующей командой: `git update-index --assume-unchanged ./volumes/workspace/dags /.airflowignore`

4.3 Правила именования DAG'ов и task'ов

- Системные DAG'и и task'и должны иметь префикс `df_`
- SubDAG'и должны иметь префикс `<id_родителя>`.
- DAG'и task'и в папке examples должны иметь префикс `df_example_`
- DAG'и task'и в папке tests должны иметь префикс `df_test_`
- DAG'и task'и в папке tests, относящиеся к проектам, должны иметь префикс `test_<имя_проекта>_`
- Task'и должны иметь уникальные id, что достигается, например, добавлением префикса с id DAG'a: `<dag_id>.<task_id>`

4.4 Операторы

Как правило, DAG'и передают в докер оператор следующие параметры, в качестве переменных окружения:

- `AF_EXECUTION_DATE`: ds
- `AF_TASK_OWNER`: task.owner
- `AF_TASK_ID`: task.id
- `AF_RUN_ID`: run_id

Дополнительные, часто используемые, параметры:

- `AF_PRODUCER`: путь к описанию источника/ов данных, например, файла, относительно папки метаданных
- `AF_CONSUMER`: путь к описанию получателя/ей данных, например, таблицы, относительно папки метаданных

- `AF_SCRIPT_PATH`: путь к директории с SQL скриптами или к непосредственно к файлу
- `AF_NOCOUNT`: флаг, определяющий необходимость выполнять `SET NOCOUNT ON` перед выполнением sql-скриптов в SQL Server. True - значение по умолчанию, т.е. `SET NOCOUNT ON` будет выполнено
- `AF_SCRIPT_RE`: регулярное выражения отбора файлов, если в `AF_SCRIPT_PATH` указана директория
- `AF_METADATA_PATH`: путь к директории с метаданными CDM или к непосредственно к файлу
- `AF_METADATA_RE`: регулярное выражения отбора файлов, если в `AF_METADATA_PATH` указана директория
- `AF_METADATA_DEPTH`: параметр, определяющий максимальную глубину поиска файлов метаданных.
  Если его значение равно 1, то смотрим только в `AF_METADATA_PATH` без подкаталогов.
  Если 2, то `AF_METADATA_PATH` и его вложенные каталоги.
  Если 3, то смотрим 3 уровня.
  И так далее.
  0 - значение по умолчанию, поиск в `AF_METADATA_PATH` и всех уровнях его подкаталогах
- `AF_JSON_PATH`: путь к директории с JSON наполнением таблиц
- `AF_JSON_RE`: регулярное выражения отбора файлов, если в `AF_JSON_PATH` указана директория
- `AF_PROCEDURE_PATH`: путь к SQL процедуре для запуска
- `AF_SINGLETRAN`: `default` - `False`, позволяет выполнять sql-стейтменты одного файла в одной транзакции
- `AF_IS_SELECT`: `default` - `False`, позволяет явно запросить возврат результата sql запроса типа `select`

- AF_RULES_PATH: путь к директории с правилами для загрузки в базу DWH
- AF_RULES: список правил для загрузки в базу DWH
- AF_JOB_ID: числовой идентификатор запуска таска
- AF_DWH_DB_DIALECT: диалект SQL, используемый СУБД с данными
- AF_DWH_DB_NAME: наименование БД с данными
- AF_DWH_DB_SCHEMA: схема с данными
- AF_DWH_DB_USER: пользователь для подключения к БД с данными
- AF_DWH_DB_PASSWORD: пароль пользователя для подключения к БД с данными
- AF_DWH_DB_SERVER: сервер СУБД
- AF_DWH_DB_PORT: порт сервера СУБД
- AF_LOGLEVEL: уровень логирования в операторе, возможные значения: 'error', 'warning', 'info', 'debug', значение по умолчанию - 'info'
- AF_S_: автоматически генерируемый параметр на основе AF_DWH_DB_SCHEMA, равен AF_DWH_DB_SCHEMA+., либо '', если AF_DWH_DB_SCHEMA не задана
- AF_SC_: автоматически генерируемый параметр на основе AF_DWH_DB_SCHEMA, равен AF_DWH_DB_SCHEMA+.-<имя_базы>, либо '', если AF_DWH_DB_SCHEMA не задана
- AF_DWH_DB_SCHEMA*: схема с данными, где * постфикс, позволяющий задавать более одной схемы, для которой автоматически генерируются соответствующие AF_S*_ и AF_SC*_
- AF_LOADRUN: флаг, определяющий небходимость добавления информации о выполняемой загрузке данных в таблицу r5_load_run

- `AF_LOADRUN_ID`: идентификатор записи из таблицы `r5_load_run` с информацией о выполняемой загрузке
- `AF_UNIT_ID`: идентификатор подразделения источника данных загрузки для таблицы `r5_load_run`
- `AF_DATE_BEGIN`: дата начала периода загрузки для таблицы `r5_load_run`
- `AF_DATE_END`: дата окончания и периода загрузки для таблицы `r5_load_run`
- `AF_DT`: дата загрузки для таблицы `r5_load_run`, значение по умолчанию - текущая дата
- `AF_TS`: время загрузки для таблицы `r5_load_run`, значение по умолчанию - текущее время
- `AF_EXTRACT_FORMULAS`: позволяет загружать формулы вместо сохраненных результатов предыдущего расчета из Excel

Каждый исполнитель должен получать тот набор параметров, который ему необходим.

Неизвестные параметры должны просто игнорироваться. Опциональные параметры, в случае их отсутствия, должны принимать значения по умолчанию.

В докер операторах можно использовать язык движка шаблонов Jinja2.

Рекомендуется ограничивать потребление ресурсов хоста docker операторами, например, используя параметры cpus, mem_limit, execution_timeout

4.5 Extract

4.5.1 Общие положения

Управляет передачей данных из источника в получатель экстрактор (extractor). В качестве источников и получателей данных используются сущности модели CDM.

Все сущности имеют общий постфикс - Entity. Для передачи данных в получатели используются аплоадеры (uploaders). Типы сущностей-получателей данных содержат в своем наименовании инфикс Local, при этом не обязательно принадлежат одному семейству.

Для получения данных из источников используются адаптеры (adapters).

Типы сущностей-источников, в зависимости от природы источника, в своем наименовании могут содержать инфиксы Web, File, но никогда Local. Таким образом, аплоадеры всегда взаимодействуют только с Local типами сущностей, а адаптеры - со всеми остальными, т.е. с внешними системами.

4.5.1.1 Аннотации сущности типа LocalEntity

Поддержка зависит от конкретного обработчика.

- `precreateEntity`, `default = False`, автоматически создавать сущность (таблицу)
- `ignoreExistingTable`, `default = True`, если включено, пропустить создание уже существующей таблицы
- `purgeExistingData`, `default = False`, очищать существующие данные перед добавлением новых, используется каскадный `truncate` для postgresql и `delete` в mssql для таблицы дополнительных свойств
- `enforceStrictMapping`, `default = True`, требовать, чтобы в сущности (таблице) присутствовали все атрибуты модели
- `schema`, `default = None`, схема, к которой принадлежит таблица в базе данных, по умолчанию используется схема из `AF_DWH_DB_SCHEMA`. Если указана схема локальной сущности, то она переопределяет схему по умолчанию, а схема ссылочной сущности переопределять схему локальной сущности
- `useBulkInsert`, `default = True`, использовать пакетную вставку при загрузке данных в таблицу БД

- `recreateIfOutOfSync`, `default = False`, пересоздавать таблицу БД в случае несоответствия ее структуры в части полей и их типов описанию, обрабатывается только при `enforceStrictMapping = True`

4.5.1.2 Поддерживаемые аннотации атрибутов сущности типа LocalEntity

- `role`, `default = None`, роль (поведение) атрибута
- `primaryKey`, `default = False`, если включено, атрибут включается в первичный ключ, если такой атрибут один и его тип integer или int64, то он создается как identity(1,1)
- `logicalKey`, `default = False`, если включено, атрибут входит в логический ключ
- `sequence`, `default = None`, наименование сиквенса, если задано и атрибут с `primaryKey = True`, то дополнительно к таблице создается объект sequence с указанным наименованием
- `sequenceCache`, `default = None`, размер кэша сиквенса (`integer`)
- `default`, `default = None`, значение по умолчанию, в текущей версии поддерживаются только строковые значения
- `nullable`, `default = True`, может ли атрибут содержать пустые (`null`) значения
- `unique`, `default = False`, если включено, значения атрибута должны быть уникальны
- `softNullable`, `default = False`, переопределяет поведение `nullable = False`, т.е. если данная аннотация включена - не создается `Not Null Constraint`
- `softUnique`, `default = False`, переопределяет поведение `unique = True`, т.е. если данная аннотация включена - не создается `Unique Constraint`

- `index`, `default = False`, если включено, по атрибуту строится индекс, есть ограничения, например, для строкового типа должна быть указана размерность
- `length`, `default = None`, размерность для типов `string`, `varchar` и `unicode`
- `format`, `default = None`, формат поля, частичная поддержка регулярных выражений
- `precision`, `default = 28`, общее максимальное количество знаков, используется для типа `decimal`
- `scale`, `default = 10`, количество знаков после запятой используется для типа `decimal`

4.5.1.3 Поддерживаемые типы данных полей

- `string`, соответствует `nvarchar(x)`
- `unicode`, соответствует `nvarchar(x)`
- `varchar`, соответствует `varchar(x)`
- `text`, соответствует `clob`
- `guid`, соответствует `uuid`
- `int64`, соответствует `integer`
- `integer`, соответствует `integer`
- `date`, соответствует дате без времени
- `dateTime`, соответствует дате со временем
- `timestamp`, соответствует дате со временем с миллисекундами
- `decimal`, соответствует числу с десятичными знаками, точность по умолчанию: `28,10`
- `boolean`, соответствует `True|False` либо `1|0`
- `float`, соответствует `float`

4.5.1.4 Свойства по умолчанию атрибутов по ролям (тип атрибута может быть переопределен заданием dataType)

- `id`: `integer, primary key`, суррогатный ключ
- `code`: `nvarchar(32), primary key`, читаемый идентификатор записи
- `rid`: `guid, primary key`, идентификатор строки
- `sid`: `varchar(128), not null`, идентификатор источника
- `runid`: `varchar(128), not null, index`, идентификатор запуска
- `jobid`: `integer, not null, index`, идентификатор задания
- `datebegin`: `datetime, not null`, дата начала периода действия
- `dateend`: `datetime, not null`, дата окончания периода действия
- `unitid`: `nvarchar(128), not null`, идентификатор структурного подразделения
- `cd`: `date, default date`, дата записи (created date)
- `cdt`: `date, default date`, дата записи (created datetime)
- `cts`: `timestamp, default utcnow`, дата и время записи (created timestamp) (подразумевается дата и время создания объекта, а не записи в техническом плане)
- `tag`: `integer`, битовая маска в десятичной системе, используется для тегирования данных

4.5.1.5 Поддерживаемые аннотации связей типа SingleKeyRelationship

- `generateKey`, `default = False`, генерация внешних ключей в БД при создании таблиц по метаданным
- `onUpdate`, поведение при изменении внешнего ключа, допустимые значения, поддержка и имплементация зависят от СУБД

- `onDelete`, поведение при удалении внешнего ключа, допустимые значения, поддержка и имплементация зависят от СУБД

4.5.1.6 Аннотации сущности типа ViTableEntity (поддержка зависит от конкретного обработчика)

- `precreateEntity`, `default = False`, автоматически создавать сущность (таблицу)
- `ignoreExistingTable`, `default = True`, если включено, пропустить создание уже существующей таблицы
- `purgeExistingData`, `default = False`, очищать существующие данные перед добавлением новых, используется каскадный `truncate` для postgresql и `delete` в mssql для таблицы дополнительных свойств
- `enforceStrictMapping`, `default = True`, требовать, чтобы в сущности (таблице) присутствовали все атрибуты модели
- `schema`, `default = None`, база, к которой принадлежит таблица, по умолчанию используется схема из `AF_DWH_DB_SCHEMA`. Если указана схема локальной сущности, то она переопределяет схему по умолчанию
- `recreateIfOutOfSync`, `default = False`, пересоздавать таблицу БД в случае несоответствия ее структуры в части полей и их типов описанию, обрабатывается только при `enforceStrictMapping = True`

4.5.1.7 Поддерживаемые аннотации атрибутов сущности типа LocalEntity

- `role`, `default = None`, роль (поведение) атрибута
- `primaryKey`, `default = False`, если включено, атрибут включается в первичный ключ, если такой атрибут имеет целочисленный тип, то он создается как `identity(1,1)`

- `sequence`, `default = None`, наименование сиквенса, если задано и атрибут с `primaryKey = True`, то дополнительно к таблице создается объект sequence с указанным наименованием

4.5.1.8 Поддерживаемые типы данных полей

- `string`, соответствует `nvarchar(x)`
- `integer`, соответствует `integer` (дополнительно поддерживаются `short`, `long`)
- `time`, соответствует времени
- `date`, соответствует дате без времени
- `dateTime`, соответствует дате со временем, при взаимодействии с API используется `'datetime'`
- `real`, соответствует `real`
- `boolean`, соответствует `true|false`
- `float`, соответствует `float`

4.5.1.9 Свойства по умолчанию атрибутов по ролям (тип атрибута может быть переопределен заданием `dataType`)

- `id`: long, primary key, суррогатный ключ, для которого автоматически создается sequence с наименованием в формате `<table_name>_<field_name>_sq`, если sequence не задан явно.

4.5.2 Extract из Excel

Поддерживаются 3 диалекта СУБД: mssql, postgresql, частично qhb (без функционала переноса данных между базами) и частично sqlite (для быстрого тестирования).

Переменные окружения будут использованы в файлах метаданных как переменные подстановки только если они имеют префикс `AF_`. Соответствие сущностей источника и получателя определяется по name. Сущности, не найденные и в источнике и получателе - игнорируются.

Соответствие атрибутов сущностей источника и получателя определяется по name.

В модели Producer'a может быть указан порядковый номер/буквенное обозначение колонки и/или ее наименование для сопоставления с именем атрибута модели Consumer.

Обязательные свойства атрибутов сущности типа `ExcelFileEntity`: `name`.

Опциональные свойства атрибутов сущности типа `ExcelFileEntity`: `ordinal/columnLetter`, `columnName`. У каждого атрибута сущности типа `ExcelFileEntity` должен присутствовать `ordinal/columnLetter` либо `columnName`. Если у сущности задана опция `columnNamesRow`, но у атрибута не задан `columnName`, значение `columnName` берется из `name`.

Значения опций в аннотациях могут быть строковыми, числовыми, а также `null`, `false`, `true`. Распознаваемые специальные значения опций, задаются в кавычках: `"null"`, `"false"`, `"true"`.
Одномоментно один экстрактор обрабатывает один файл.
В описаниях аннотаций термины "маска" и "регулярное выражение" взаимозаменяемы.
Объекты `Connection` для подключения к СУБД `QHB` должен иметь тип `Postgres` и `Extra` опцию `dialect = qhb`, если диалект не задан через переменную окружения `AF_DWH_DB_DIALECT`.

4.5.2.1 Поддерживаемые аннотации сущности типа `ExcelFileEntity`:

- `rowsNumberToSkip`, `default = 0`, пропускать первые N строк, нумерация с 1
- `rowsLoadLimit`, `default = 0`, обработать не более N строк начиная с 1, 0 - обработать все
- `rowsBatchSize`, `default = 10000`, количество строк (записей) данных, обрабатываемых как один целостный набор

- `batchSizeBreadth`, `default = 15`, максимальное количество столбцов в пачке, если оно меньше числа обрабатываемых столбцов, то пропорционально уменьшается `rowsBatchSize`
- `skipBlankRows`, `default = True`, пропускать пустые строки
- `rowMaskToSkip`, маска пропуска строк
- `rowMaskToSkipL` - `rowMaskToSkip` по склеенным значениям ячеек
- `headLastRowMask`, маска последней строки шапки для пропуска, не может идти позже строки с наименованиями `columnNamesRow`
- `skipHeadLastRow`, `default = True`, пропускать последнюю строку шапки
- `headLastRowMaskStrict`, `default = False`, искать маску последней строки шапки только в обрабатываемых столбцах
- `headLastRowMaskL` - `headLastRowMask` по склеенным значениям ячеек
- `minHeadRow`, `default = 1`, определяет номер первой строки шапки
- `minHeadRowMask`, `default = None`, маска первой строки шапки, переопределяет `minHeadRow`, если срабатывает
- `maxHeadRowOffset`, `default = 20`, определяет максимально возможное количество строк в шапке
- `minHeadRowMaskL` - `minHeadRowMask` по склеенным значениям ячеек
- `tailFirstRowMask`, маска первой строки подвала для пропуска
- `skipTailFirstRow`, `default = True`, пропускать первую строку подвала
- `tailFirstRowMaskL` - `tailFirstRowMask` по склеенным значениям ячеек

- `detectMergedCells`, `default = False`, определять объединенные ячейки и заполнять их, включение данной настройки замедляет обработку
- `enforceStrictMapping`, `default = True`, требовать, чтобы в источнике были определены все атрибуты получателя
- `lowMemoryMode`, `default = False`, не загружать файл целиком в память (время загрузки значительно возрастет, использовать только для очень больших файлов), поддерживается только для `xlsx`
- `ignoreAbsentFiles`, `default = False`, игнорировать отсутствие файла для загрузки, завершением работы без ошибки
- `sheetOrdinal`, `default = 1`, номер загружаемого листа Excel, нумерация с 1, параметр игнорируется, если задана опция `sheetName`
- `sheetName`, `default = None`, имя загружаемого листа Excel
- `reInSheetNames`, `default = False`, идентифицировать листы Excel по регулярному выражению, если sheetName не задан - игнорируется
- `ignoreAbsentSheets`, `default = False`, пропускать файл и продолжать работу, даже если искомый лист не найден
- `columnNamesRow`, `default = 0`, номер строки с наименованиями, идентифицирующими атрибут получателя, нумерация с 1, должна идти позже шапки `headLastRowMask`
- `columnNamesRowMask`, `default = None`, маска для поиска строки с наименованиями, идентифицирующими атрибут получателя
- `columnNamesRowMaskL` - `columnNamesRowMask` по склеенным значениям ячеек
- `reInColumnNames`, `default = False`, идентифицировать колонки по регулярному выражению

- `dateBeginVariable`, `default = None`, переменная (Variable), содержащая дату начала периода загрузки в формате `YYYYMMDD`, загрузка выполняется в строку
- `dateEndVariable`, `default = None`, переменная (Variable), содержащая дату окончания периода загрузки в формате `YYYYMMDD`, загрузка выполняется в строку
- `unitIdVariable`, `default = None`, переменная (Variable), содержащая идентификатор предприятия, по которому загружаются данные
- `reInPathDelimiter`, `default = None`, составляющие полного пути к файлу, обернутые в разделитель, если он задан, обрабатываются как регулярное выражение
- `dateBeginGroup`, `default = None`, номер регулярного выражения и номер группы в нем в формате `'нрв:нг'`, как дата начала периода загрузки в пути к файлу, если `reInPathDelimiter` `is not None` и не определена `dateBeginVariable`, формат даты в пути - `YYYYMMDD`
- `dateEndGroup`, `default = None`, номер регулярного выражения и номер группы в нем в формате 'нрв:нг', как дата окончания периода загрузки в пути к файлу, если `reInPathDelimiter` `is not None` и не определена `dateEndVariable`, формат даты в пути - `YYYYMMDD`
- `unitIdGroup`, `default = None`, номер регулярного выражения и номер группы в нем в формате `'нрв:нг'`, как идентификатор предприятия в пути к файлу, если `reInPathDelimiter` `is not None` и не определена `unitIdVariable`
- `archiveFolder`, `default = None`, директория для переноса обработанных файлов с сохранением структуры с первой переменной папки, если такая есть

- `failedFolder`, `default = None`, директория для переноса обработанных с ошибками файлов, структура каталогов с файлами сохраняется, если путь к файлам содержит переменные
- `ignoreArchiveErrors`, `default = False`, игнорировать ошибки при архивации
- `removeArchivedFolders`, `default = False`, удалять пустые переменные папки, обрабатывается только если задана `archiveFolder`
- `doNotPipeSkipped`, `default = False`, не выполнять постобработку файлов, пропущенных по каким-либо причинам
- `skipHiddenSheets`, `default = True`, не обрабатывать скрытые листы, отключение данной настройки замедляет обработку, поддерживается только `xlsx` и `xls`
- `skipHiddenParts`, `default = True`, не обрабатывать скрытые строки и столбцы, поддерживается только `xlsx` и `xls`
- `skipHiddenRows`, `default = False`, не обрабатывать скрытые строки (учитывается только в режиме расширенной загрузки), поддерживается только `xlsx`
- `skipHiddenColumns`, `default = False`, не обрабатывать скрытые столбцы (учитывается только в режиме расширенной загрузки), поддерживается только `xlsx`
- `loadExtra`, `default = False`, расширенная загрузка, включение данной настройки позволяет загружать форматирование ячеек и переключает загрузчик на обработку только `xlsx`
- `readonlyExtra`, `default = True`, при расширенной загрузке открывать файл только для чтения (влияет на загружаемые свойства, например, отсутствует `col_idx`, форматирование объединенных ячеек не дублируется, а их индивидуальные значения считываются)

- `maxBlanksToProcess`, `default = None`, обрабатывать максимум пустых строк подряд, для случаев, когда библиотека не может определить конец файла, 0 - завершать обработку на первой пустой строке
- `ignoreBogusFiles`, `default = False`, пропускать файлы, содержащие структурные ошибки
- `multiAccessMode`, `default = False`, позволяет одновременный доступ на запись к файлу нескольких процессов
- `skipMappingErrors`, `default = False`, позволяет пропускать файлы с ненайденными столбцами без падения
- `loadMultipleColumns`, `default = False`, позволяет загружать несколько столбцов в один атрибут
- `loadMultipleSheets`, `default = False`, позволяет загружать все листы подходящие под `sheetName` при `reInSheetNames = True` в одну сущность
- `convertToXLSX`, `default = False`, конвертирует файл в `xlsx` перед экстрактом, с целью повышения совместимости с расширенными функциями обработки данных, а также типов извлекаемых данных, поддерживаемые форматы: `'xls'`, `'xlsb'`, `'ods'`

4.5.2.2 Поддерживаемые аннотации атрибутов сущности типа ExcelFileEntity

- `extraToLoad`, `default = None`, перечень дополнительных свойств ячеек, подлежащих загрузке, значения загружаются как строки, формат перечня: `свойство.один:свойство.два:...:свойство.эн`, включается при `loadExtra = True`
- `isRow`, `default = False`, является ли источником значений атрибута строка, строки не могут располагаться ниже строки, заданной `columnNamesRow` или `columnNamesRowMask`

- `rowColumn`, `default = 0`, ограничивает чтение строки заданным номером колонки, используется для загрузки значения определенной ячейки из строки при `isRow = true`
- `rowColumnMask`, `default = None`, определяет номер колонки `rowColumn` по маске, имеет приоритет над `rowColumn`
- `rowChainMask`, `default = None`, определяет номер колонки `rowColumn` по маске, используя значения всех ячеек начиная со строки по `firstRowMask`, имеет приоритет над `rowColumnMask`
- `firstRowMask`, `default = None`, регулярное выражение, определяющее строку, с которой начинается отсчет строк, найденная строка имеет `ordinal = 1`
- `targetAttribute`, `default = None`, имя атрибута при `isRow = true` и `rowColumn = 0`, для которого определяется значение
- `const`, `default = None`, загрузка константны в атрибут при `isRow = false`, поддерживает специальные значения
- `valueMask`, `default = None`, регулярное выражение, определяющее значение из `rowColumn` или `const` записывается значение группы 0 или 1, по умолчанию используется `^(.*)$`
- `setColumnOrdinal`, `default = False`, переопределяет номер колонки связанного атрибута `targetAttribute` на колонку, определенную в `rowColumn`, `rowColumnMask` или `rowChainMask`
- `doNotLoad`, `default = False`, при `isRow = true` позволяет не загружать атрибут в таблицу, метаданные получателя при этом не проверяются

4.5.2.3 Специальные значения аннотации `const` атрибутов сущности типа `ExcelFileEntity`:

- `sheetName` - значение аннотации `const` приравнивается к наименованию загружаемого листа, если загрузка выполняется по наименованию листа, иначе - к номеру листа

- `fileName` - имя файла
- `filePath` - полный путь файла

Если `ordinal` для атрибута типа `const = 0`, то при отсутствии какого-либо значения для этого атрибута, загрузка завершится с ошибкой

Если `ordinal` для атрибута типа `const != 1`, то при отсутствии какого-либо значения для этого атрибута будет загружено значение из соответствующего атрибута, если он существует, иначе загрузка завершится с ошибкой

4.5.2.4 Примеры дополнительных свойств ячеек для аннотации extraToLoad:

- `row` - номер строки, например, `'1'`
- `column` - номер колонки, например, `'1'`
- `coordinate` - координаты строки, например, `'E7'`
- `font.b` - признак жирного шрифта (`'True'|'False'`)
- `font.i` - признак курсива (`'True'|'False'`)
- `alignment.indent` - размер отступа, например, `'1.0'`
- `fill.bgColor.rgb` - цвет фона, например, `'FFCCFFCC'`

4.5.2.5 Специальные дополнительные свойства аннотации extraToLoad:

- `range` - диапазон объединения ячейки (при наличии), пример - `'A1:C3'`
- `sheet.hidden` - признак скрытия листа (`'True'|'False'`)
- `row.hidden` - признак скрытия строки (`'True'|'False'`)
- `column.hidden` - признак скрытия колонки (`'True'|'False'`)
- `row.outline` - уровень группировки строки например, `'0'`
- `column.outline` - уровень группировки строки например, `'0'`

Пример значения аннотации `extraToLoad: "font.b:font.i:alignment.indent:fill.bgColor.rgb:range"`

Значения дополнительных свойств аннотации `extraToLoad` хранятся в автоматически создаваемой таблице с наименованием `<имяТаблицыСДанными>_f`.

Структура таблицы: `id [int identity not null], rowId [int fk not null], column [varchar(128) not null], key [varchar(128) not null], value [varchar(256) not null]`

Для включения загрузки дополнительных свойств необходимо выполнить следующие шаги:

- в аннотации источника включить loadExtra
- в аннотации атрибутов источника задать значение extraToLoad
- в таблицу данных добавить атрибут типа uuid [not null, unique], в метаданных получателя задать этому атрибуту роль rid
  Поле типа uuid в таблице данных необходимо для привязки дополнительных свойств к строкам данных до непосредственной вставки строк данных в таблицу, т.к. id на этот момент еще неизвестен.
  В результате загрузки, помимо таблицы данных будет заполнена таблица с дополнительными свойствами <имя_таблицы_данных>_f, с привязкой по rowid к строкам таблицы данных и свойствами в виде пар ключ-значение, в разрезе полей.
  Пример скрипта добавления поля типа uuid в таблицу данных (postgres):

```
CREATE EXTENSION IF NOT EXISTS "uuid-ossp";

alter table "public"."test_formatting" add "gid2" uuid not null default uuid_generate_v1();

alter table "public"."test_formatting" add CONSTRAINT "test_formatting_gid2_key" UNIQUE ("gid2");

alter table "public"."test_formatting" drop CONSTRAINT "test_formatting_gid2_key";

alter table "public"."test_formatting" drop column "gid2";
```

4.5.2.6 Дополнительные разъяснения по загрузке

Сущность обрабатывается без ошибки загрузки, даже если набор атрибутов источника и получателя отличается, если в метаданных и источника и получателя опция `enforceStrictMapping` явно отключена.

При определении номера и группы регулярного выражения в формате `'нрв:нг'`, регулярные выражения нумеруются с 1, а группы с 0, где группа 0 - значение выражения целиком.

Если при загрузке сложной шапки возникает ошибка `Ошибка экстрактора Excel: Столбца для <атрибут> нет в <файл>, метаданные не соответствуют файлу?`, возможно, что загрузчик не может определить границу шапки и данных, в этом случае следует задать `columnNamesRow` или ее аналог более точно, чтобы опция срабатывала только точно на нужной строке.

Если при загрузке возникает ошибка `Не задан номер или буквенное обозначение колонки для атрибута <имя>`, следует убедиться, что у соответствующего атрибута указан `ordinal/columnLetter`.

Если при загрузке возникает ошибка `Коллизия колонки <имя или маска> (номер) и атрибута <имя>.`, вероятно произошло совпадение наименований колонок данных, либо маска наименования подошла более чем одному значению.

Экстракт из `xls`, `xlsb`, `ods`, `xlsm` является эспериментальным и обладает рядом ограничений, так, в частности:

- ни для одного из указанных типов файлов не поддерживаются параллельный экстракт и экстракт форматирования, для типов `xlsb`, `ods` не поддерживаются идентификация объединенных ячеек, для типов `xlsm`, `xlsb` не поддерживается экстракт из скрытых ячеек, а для `ods` экстракт из скрытых ячеек выполняется всегда
- при экстракции из `xls`, `xlsm`, `xlsb`, `ods` файлов и включенной аннотации `convertToXLSX`, либо при включенной параллельной

экстракции (`multiAccessMode = True`), в контейнере оператора должно быть достаточно сводного места (hd) для полной копии обрабатываемого файла

- типы данных в результате экстракции из `xls`, `xlsm`, `xlsb`, `ods` могут различаться при различных состояниях аннотации `convertToXLSX`
- типы данных при экстракте из `xlsb` без конвертации идентифицируются только как строковые и числовые
- при экстракции из `xlsb`, `ods` пустые строки до начала данных игнорируются

4.5.3 Extract из XML API NetDB

Используется Excel экстрактор, в части не касающейся обработки форматирования, формул, буквенных идентификаторов и листов, ввиду отсутствия поддержки форматом таковых.

Для описания источника используется тип сущности `NetDBFileEntity`

4.5.3.1 Уникальные аннотации сущности типа NetDBFileEntity:

- `spreadMergedRows`, `default = False`, дублировать значения в объединенных строках `rowspan`
- `spreadMergedColumns`, `default = False`, дублировать значения в объединенных колонках colspan

4.5.3.2 Уникальные аннотации атрибутов сущности типа NetDBFileEntity:

- `param`, `default = None`, загрузка параметров из элемента `<params />` в атрибут при `isRow = false`, формат переменной параметров - `@ndbparam<индекс_параметра_с_1>`

4.5.4 Extract из CSV

Используется Excel экстрактор, в части не касающейся обработки форматирования, формул и буквенных идентификаторов, ввиду отсутствия поддержки форматом таковых.

Для описания источника используется тип сущности `CSVFileEntity`

Мультилистовые CSV файлы в текущей версии не поддерживаются

4.5.4.1 Уникальные аннотации сущности типа CSVFileEntity:

- `encoding`, `default = utf-8`, кодировка файла
- `delimiter`, `default = ,`, разделитель атрибутов
- `detectFloat`, `default = False`, пытаться распознать дробные числа, слегка замедляет обработку
- `detectInt`, `default = False`, пытаться распознать целые числа, слегка замедляет обработку, действует только при включенной detectFloat
- `detectDatetime`, `default = False`, пытаться распознать даты, слегка замедляет обработку

4.5.5 Extract из PIPware API

Используется Excel экстрактор без возможности разбора шапки таблицы, ввиду отсутствия ее поддержки форматом.

Для описания источника используется тип сущности `PIPwareEntity`

4.5.5.1 Уникальные аннотации сущности типа PIPwareEntity:

`rowsNumberToSkip`, `default = 1` (наименования колонок), пропускать первые N строк, нумерация с 1.

4.5.6 Extract из KZStat API

Используется Excel экстрактор,в части не касающейся обработки форматирования, формул и буквенных идентификаторов, ввиду отсутствия поддержки форматом таковых.

Для описания источника используется тип сущности `KZStatWebEntity`

4.5.7 Extract из KZTask API (API Поручений)

Используется Excel экстрактор,в части не касающейся обработки форматирования, формул и буквенных идентификаторов, ввиду отсутствия поддержки форматом таковых. Для описания источника используется тип сущности `KZTaskWebEntity`. Идентификация атрибутов выполняется по наименованиям (`columnName`). Адаптер поддерживает следующие `extra` опции объекта `Connection`:

- `auth_endpoint` - адрес авторизации относительно API entry point (`Connection.host`). По умолчанию задан `login`. Обязательно наличие логина и пароля в объекте `Connection` для авторизации.

4.5.8 Extract из Visiology DC API

Используется Excel экстрактор в качестве обработчика измерения как тела таблицы.

Для описания источника используется тип сущности `DCDimensionEntity`
Идентификация атрибутов выполняется по наименованиям (`columnName`), либо в порядке их идентификаторов (`ordinal`).

Порядок следования колонок строк: `id`, `name`, `path`, `attr1 value, .., attrN value` Количество записей, получаемое через API в одном запросе можно ограничить как значением ключа `'batch'` в `'extra'` подключения, так и аннотацией `dcBatchSize` метаданных источника

4.5.8.1 Уникальные аннотации сущности типа `DCDimensionEntity`:

- `dcBatchSize`, `default = 1000`, максимальное количество элементов измерения в одном ответе API
- `pathDelimiter`, `default = /`, задает разделитель элементов `path` атрибутов

- `rowsNumberToSkip`, `default = 1` (наименования колонок), пропускать первые N строк, нумерация с 1.

4.5.9 Extract из БД Postgres, MS SQL Server, Firebird, Oracle (совместимость драйвера - 12.1)

Выполняется отдельным раннером `/app/source/python/run_sql_move.py`.

Для получения загружаемого набора данных из таблицы в базе источнике должен быть определен SQL скрипт по пути `AF_PRODUCER_SCRIPT_PATH`.

Для записи данных в таблицу базы получателя должен быть определен SQL скрипт по пути `AF_CONSUMER_SCRIPT_PATH`.

Дополнительно использует переменные окружения `AF_DWH_DB_*_PRODUCER` и `AF_DWH_DB_*_CONSUMER` для подключения к источнику и получателю, соответственно.

Количество записей, отправляемых в таблицу-получатель за раз, ограничивается переменной окружения `AF_BATCH_SIZE`, по умолчанию - 1000, актуально только для получателя Postgres.

Для обработки больших объемов данных, например - превышающих доступную контейнеру оперативную память, следует ограничивать предельное количество записей, передаваемых единовременно, через переменную `AF_PRODUCER_BATCH_SIZE`.

`Schema` объекта Connection типа Oracle определеяет Oracle Instance.

Объект Connection для подключения к СУБД Firebird должен иметь тип `File (path)` и `Extra` опцию `dialect = firebird`. Connection `Schema` определеяет `Database path`.

4.5.10 Extract из БД Postgres, MS SQL Server в Visiology DataCollect через DC API

Выполняется отдельным раннером `/app/source/python/run_sql_to_viapi.py`.

Позволяет выполнять загрузку результатов SQL запросов в DC. Поддерживаются методы `post`, `put` и `delete` для измерений. Поддерживаются следующие переменные окружения:

`AF_DWH_DB_CONNECTION_PRODUCER` и `AF_DWH_DB_CONNECTION_CONSUMER` для подключения к источнику и получателю

`AF_BATCH_SIZE` - количество элементов, единовременно отправляемых на эндпойнт, по умолчанию - 1000

`AF_SCRIPT_PATH` - путь к шаблону SQL select источника

`AF_DELIMITER`- разделитель для элементов `path`, по умолчанию - `/`

`AF_OPERATION` - тип операции (`create`, `delete`, `update`), по умолчанию `create`

`AF_ENDPOINT` - API endpoint

Структура запроса для загрузки данных в измерение:

```
select element_id, element_name, element/path, attr1_id, attr1_value, ..., attrN_id, attrN_value
```

`element_id`, `element_name`, `element/path` - обязательны, если `element/path` пуст, то следует передавать `''`

Структура запроса для удаления элементов измерения по `Id`:

```
select id as value, 'Id' as type, 'условие' as condition, '' as name
```

Структура запроса для обновления элементов измерения по `name`:

```
select 'наименование элемента' as value, 'name' as type, 'условие' as condition, '' as name, 'новое значение' as n_value, 'attribute' as n_type, 'наименование атрибута' as n_name
```

4.5.11 Extract из XML файлов Электронного бюджета

Используется Excel экстрактор, в части не касающейся обработки форматирования, формул, буквенных идентификаторов и листов, ввиду отсутствия поддержки форматом таковых.

Для описания источника используется тип сущности `EBFileEntity`. Идентификация атрибутов выполняется по наименованиям (`columnName`), либо в порядке их идентификаторов (`ordinal`). Отчетный период данных содержится в имени XML файла, для извлечения его оттуда необходимо

использовать аннотации `reInPathDelimiter` и одну из `dateBeginGroup` или `dateEndGroup`.

4.5.12 Extract из XML файлов Электронного учета МинСельХоза

Используется Excel экстрактор, в части не касающейся обработки форматирования, формул, буквенных идентификаторов и листов, ввиду отсутствия поддержки форматом таковых.

Для описания источника используется тип сущности `MSHEUFileEntity`. Идентификация атрибутов выполняется по наименованиям (`columnName`).

4.5.13 Extract из Excel 2003 XML Spreadsheet

Используется Excel экстрактор, в части не касающейся обработки форматирования, формул, буквенных идентификаторов и листов, ввиду отсутствия поддержки форматом таковых.

Для описания источника используется тип сущности `Excel2003XMLFileEntity`

4.5.14 Extract из API Электронного бюджета

Используется Excel экстрактор, в части не касающейся обработки форматирования, формул, буквенных идентификаторов и листов, ввиду отсутствия поддержки форматом таковых.

Для описания источника используется тип сущности `EBWebEntity`. Идентификация атрибутов выполняется по наименованиям (`columnName`).

4.5.14.1 Уникальные аннотации сущности типа `EBWebEntity`:

- `nullMappingErrors`, `default = False`, позволяет в случае отсутствия в источнике колонки записывать `NULL` в таблицу.

4.5.15 Extract из API, возвращающего данные в виде списка словарей в формате JSON

Используется Excel экстрактор, в части не касающейся обработки форматирования, формул, буквенных идентификаторов и листов, ввиду отсутствия поддержки форматом таковых.

Для описания источника используется тип сущности `JSONMapWebEntity`.

Идентификация атрибутов выполняется по наименованиям (`columnName`).

Адаптер поддерживает следующие `extra` опции объекта `Connection`:

- `auth - basic` или `ntlm` как метод аутентификации API
- `domain` - имя домена при `auth = ntlm`, если требуется

4.5.16 Extract из API, возвращающего данные в виде словаря массивов в формате JSON

Используется Excel экстрактор, в части не касающейся обработки форматирования, формул, буквенных идентификаторов и листов, ввиду отсутствия поддержки форматом таковых.

Для описания источника используется тип сущности `JSONListWebEntity`.

Идентификация атрибутов выполняется по наименованиям (`columnName`).

Адаптер поддерживает следующие `extra` опции объекта `Connection`:

- `auth - basic` или `ntlm` как метод аутентификации API
- `domain` - имя домена при `auth = ntlm`, если требуется
- `columns_name` - параметр ответа API содержащий массив заголовков. По умолчанию задан `columns`.
- `data_name` - параметр ответа API содержащий список массивов данных соответствующих заголовкам указанным в заголовках. По умолчанию задан `data`.

4.6 Loaders (Загрузчики)

Инструменты загрузки/кэширования (loaders) данных из различных источников.

Как правило, данные загружаются в виде файлов, которые в дальнейшем обрабатываются соответствующими экстракторами. Корневая системная директория для загруженных/закэшированных файлов: `/app/cache` (том с обязательным правом на запись, но без права на исполнение)

Системные раннер: `/app/source/python/run_loader.py` . Для подключения к провайдеру данных необходимо создать объект `Connection`, с типом подключения, определяемым конкретным загрузчиком, и указать его атрибуты. Поддерживаемые `Extra` опции подключения, задаются в виде JSON словаря:

- `run` - режим запуска, `default = `wait'`; поддерживается:
- `once` - однократная проверка с последующей загрузкой, если файлы найдены,
- `wait` - ждать пока файлы не появятся и завершить работу после их загрузки
- `repeat` - возобновить поиск после загрузки
- `pattern` - маска поиска файла (regular expression match), `default = ^(.*)$`
- `interval` - интервал в секундах между проверками наличия файлов по маске, `default = 300`
- `timeout` - таймаут ожидания появления файлов, соответствующих маске, в секундах, `default = 86400`
- `mode` - режим загрузки, `default = copy`:
- `empty` - очистить локальную папку перед загрузкой
- `copy` - загружать файл

- `cut` - загружать и удалять файл с удаленного контейнера (директории)
- `unpack` - распаковать загруженные архивы
- `delete` - удалить загруженные файлы Данные опции являются общими для всех загрузчиков.

4.6.1 Loader из облака DWASH

Минимальная, поддерживаемая, версия Dwash - 0.0.6. Взаимодействует с API Dwash, предоставляя следующий функционал:

- поиск файлов в бакете облака в соответствии с переданной маской
- загрузка найденных файлов в локальный кэш
- опциональное удаление найденных файлов из облака после их успешной загрузки
- возможность как беспрерывного мониторинга облака, так и завершения работы после загрузки найденных в рамках текущего запуска файлов.

Для подключения к провайдеру данных необходимо создать объект `Connection`.

Атрибуты подключения:

- `Conn Id` - уникальный в пределах инстанса идентификатор подключения, например - `dwash_local`
- `Conn Type` - тип подключения, как правило - `HTTP`
- `Host` - базовый url API облака, например - `https://example.com/graphql`
- `Port` - порт подключения, например - `44390`, для `https` в `Host` по умолчанияю - `443`
- `Extra` - дополнительные опции подключения к провайдеру Поддерживаемые `Extra` опции подключения, задаются в виде JSON словаря:

- `provider` - код провайдера (известного лоадера), например - `dwash`
- `mode` - режим загрузки, поддерживаются все общие режимы, указанные для загрузчиков
- `token` - токен подключения к облаку, переопределяет `Connection.Password`, который ограничен 255 символами, выдается провайдером
- `options` - дополнительные параметры конкретного загрузчика, в частности:
- `ssl_verify` - `true/false` проверка SSL сертификата, по умолчанию `true`
- `container` - контейнер (директория) поиска файлов в корзине облака, заданной токеном, `default = ''` (корень бакета)
- `descent` - загружать структуру хранения файла, по умолчанию 1 (только файл), 0 - все уровни (вышестоящие директории).
- Все опции могут быть переопределены переменными окружения таска с добавлением префикса `AF_LOADER_`. Поддерживаемые переменные окружения таска (помимо переопределяющих опции подключения):
- `AF_PROVIDER_CONNECTION` - Conn Id подключения к облаку, например - `dwash_local`
- `AF_TARGET_STORAGE` - директория сохранения/кэширования загруженных из облака файлов, по умолчанию - `<task_id>/<run_id>` относительно `/app/cache`.

4.6.2 Loader из FTP

Предоставляет следующий функционал:

- поиск файлов на FTP в соответствии с переданной маской
- загрузка найденных файлов в локальный кэш

- опциональное удаление найденных файлов с FTP после их успешной загрузки
- возможность как беспрерывного мониторинга FTP, так и завершения работы после загрузки найденных в рамках текущего запуска файлов
- Для подключения к провайдеру данных необходимо создать объект `Connection`.

  Атрибуты подключения:
- `Conn Id` - уникальный в пределах инстанса идентификатор подключения, например - `ftp_local`
- `Conn Type` - тип подключения, как правило - `FTP`
- `Host` - доменное имя / ip FTP, например - `0.0.0.0`
- `Port` - порт подключения, например - `11021`, по умолчанияю - `21`
- `login` - логин пользователя
- `Password` - пароль пользователя
- `Extra` - дополнительные опции подключения к провайдеру Поддерживаемые `Extra` опции подключения, задаются в виде JSON словаря:
  - `provider` - код провайдера (известного лоадера), например - `ftp`
  - `type` - тип подключения: FTP или FTPS: , `default = FTP`
- `level` - максимальный уровень дерева каталогов, в которых будет производиться поиск (None, 0 - нет ограничений, 1 - корневой каталог), `default = None`
- `min_level` - минимальный уровень дерева каталогов для скачивания файлов (None, 0, 1 - корневой каталог), `default = 1`
- `time_shift` - значение сдвига времени, измеряемое в секундах. Сдвиг времени - различие между местным временем сервера (FTP) и местным временем клиента (сервер polyflow) в данный

момент, т.е. по определению: `time_shift = server_time - client_time`. Например, для случая, когда на FTP-сервере `UTC+05:00`, а на сервере polyflow - `UTC+00:00`, значение будет равным `18000`

- `filter` - опциональный параметр. Строка, содержащая путь до jinja-шаблона с python-кодом и имя функции, указанное через `::`, например - `/app/source/python/scripts/custom/examples/filter_files.py j::filter_by_date`. В шаблоне доступны локальные переменные и переменные окружения. Заданная функция будет использована для фильтрации загружаемых файлов: должна возвращать True для файлов, которые должны быть скачаны. На вход функция принимает объект, поля которого являются свойствами файла:
  - `name` - имя файла (например, `name: 'file_cccc'`)
  - `path` - путь до файла, относительно рабочей директории (например, `path: 'dir_2/dir_22/dir_221/file_cccc'`)
  - `stat` - стандартный объект типа `os.path`, содержащий информацию о файле: размер, дата изменения и т.п. (например, `stat: StatResult(st_mode=33188, st_ino=None, st_dev=None, st_nlink=1, st_uid='root', st_gid='root', st_size=0, st_atime=None, st_mtime=1635435480.0, st_ctime=None))`
- `mode` - режим загрузки; дополнительно к общим для всех загрузчиков режимам, поддерживаются:
- `multipattern` - передача списка наименований (масок) файлов, поиск которых осуществляется относительно `container`
- `multicontainer` - передача списка контейнеров (рабочих директорий), относительно которых выполняется поиск по `pattern`

- `options` - дополнительные параметры конкретного загрузчика, в частности:
- `container` - контейнер (директория) поиска файлов на FTP, `default = ''` (домашняя директория, указанная в настройках FTP).
- Одновременная передача списка файлов (масок) и списка контейнеров (директорий) недопустима. Все опции могут быть переопределены переменными окружения таска с добавлением префикса `AF_LOADER_`

  Поддерживаемые переменные окружения таска (помимо переопределяющих опции подключения):
- `AF_PROVIDER_CONNECTION` - `Conn Id` подключения к облаку, например - `ftp_local`
- `AF_TARGET_STORAGE` - директория сохранения/кэширования загруженных с FTP файлов, по умолчанию - `<task_id>/<run_id>` относительно `/app/cache`

4.7 Описание базовых системных объектов продукта

4.7.1 Хуки

4.7.1.1 VisiologyHook

Подключение и низкоуровневое взаимодействие с API Visiology. Использует Connection к Visiology.

4.7.1.2 NetDBHook

Подключение и низкоуровневое взаимодействие с API NetDB. Использует Connection к NetDB.

4.7.2 Операторы

4.7.2.1 VisiologyAPIOperator

Предназначен для обращения к API Visiology. Использует VisiologyHook.

Позволяет выполнять произвольные запросы к API viqube, dc, viqube admin.

4.7.2.2 NetDBAPIOperator

Предназначен для обращения к API NetDB. Использует NetDBHook

4.7.3 Сенсоры

4.7.3.1 LocalFileSensor

Проверяет наличие объектов по указанному пути, поддерживает фильтрацию регулярными выражениями.

Обязательные аргументы сенсора:

- `path` - путь поиска, как правило, относительно `/app/share`

Опциональные аргументы сенсора:

- `pattern` - маска (регулярное выражение) поиска (`default: ^.+$`)
- `trigger_delay` - отложенное срабатывание (сек., `default: 0`)
- `files_only` - срабатывать только на файлы (`boolean`, `default: True`)
- `match_name` - сверять только наименование объекта, а не пути (`boolean`, `default: True`)
- `lazy` - срабатывать при первом совпадении или искать все (`boolean`, `default: True`)
- `do_xcom_push` - возвращать найденные объекты как xcom (`boolean`, `default: False`), стандартный аргумент с инвертированным умолчанием, при `lazy = True` возвращается путь как строка, иначе - список путей

- `depth` - максимальная глубина вложенности (`int`, `default: 1`, при 0 - глубина не ограничена), если не равен 1, автоматически включает `do_xcom_push` и отключает `lazy`, если они не заданы.

4.7.4 Соединения

4.7.4.1 Visiology

Conn Type: HTTP.

Требует наличия Host, Login, Password.

Поддерживает Extra параметры: `unit`, `api_version`, `ssl_verify`.

Параметр `unit` обязателен и может принимать следующие значения: `viqube`, `dc`.

4.7.4.2 NetDB

Conn Type: HTTP.

Требует наличия Host, а также:

- либо Login и Password для подключения с использованием `sessionid`
- либо `{ "api_key": "your-netdb-api-key" }` в Extra для подключения через `Netdb-Api-Key` (приоритетный вариант) Поддерживает Extra параметры: `api_key`, `ssl_verify`, `api_version`.

4.8 Расположение управляющих системных объектов продукта

4.8.1 DAG'и

- `dags/system/backup_environment` - бэкап папок, потенциально содержащих проектные файлы
- `dags/system/log_cleanup` - периодическая очистка логов
- `dags/system/upgrade_datadb` - создание и обновление системных объектов
- `dags/system/populate_datadb` - заполнение системных таблиц данными

4.8.2 Основные исполняемые файлы

- `source/excel/run.py` - экстракт данных из Excel файлов
- `source/sql/run.py` - выполнение SQL скриптов
- `source/sql/run_statement.py` - выполнение SQL скриптов по выражениям
- `source/sql/run_create_entity.py` - создание таблиц в БД
- `source/sql/run_procedure.py` - выполнение процедуры с возможностью предварительной загрузки метаданных в контекст
- `source/system/maintaindb/merge_data_json.py` - merge данных в таблицы БД

4.9 Создание и поддержка объектов БД

DDL операции выполняются с использованием alembic. Штатный запуск alembic выполняется DAG'ами, например, `df_upgrade_datadb`.

Для ручного запуска alembic с хоста необходимо прописать строку подключения sqlalchemy.url в соответствующем alembic.ini.

5 Аварийные ситуации

Для Системы определены следующие режимы функционирования:

- штатный;
- аварийный.

Аварийный режим функционирования Системы используется при отказе одного или нескольких компонент программного и (или) технического обеспечения.

При переходе в аварийный режим в Системе предусмотрено формирование соответствующего информационного сообщения.

После выдачи сообщения, администратору необходимо выполнить комплекс мероприятий по устранению причины перехода Системы в аварийный режим.

При работе с АИС могут возникнуть следующие неисправности, приводящие к аварийным ситуациям:

- Превышение нагрузки на АИС. В этом случае необходимо ограничить количество тяжело-нагруженных процессов или общее их количество;
- Недостаток свободной оперативной памяти на сервере. В этом случае необходимо ограничить ресурсы для контейнера.
- Другие неисправности. В случае нарушения технологического процесса или при длительных отказах технических средств администратор системы обязан сообщить о возникшей проблеме в службу технической поддержки, провести диагностику работы Системы, определить вероятную причину неисправности и передать лог-файлы из соответствующего docker-контейнера. Чтобы связаться с службой поддержки необходимо сообщить о возникшей неисправности по электронному адресу: support@polyanalitika.ru.

6 Рекомендации по освоению

Основным источником информации, используемым при освоении Системы, является данное руководство.

Начинать работу с Системой следует со знакомства с разделами руководства «Подготовка к работе», «Описание операций».

Для обеспечения успешной работы пользователям необходимо обладать основными навыками работы с веб-приложениями, опубликованными в сети Интернет.

Лист регистрации изменений

Изм.	*Номера листов (страниц)*				*Всего листов (страниц) в докумен-те*	*Номер докумен-та*	*Входящий номер сопроводи-тельного документа и дата*	*Подпись*	*Дата*
	измененных	*заменен-ных*	*новых*	*аннулиро-ванных*					

Изм.	*Лист*	*№ докум.*	*Подп.*	*Дата*	*.РЭ*	*Лист* 56

Подпись и дата | *Инв. № дубл.* | *Взам. инв. №* | *Подпись и дата* | *Инв. № подл.*

Подпись и дата

Инв. № дубл.

Взам. инв. №

Подпись и дата

Инв. № подл.

					.РЭ	Лист
						57
Изм.	Лист	№ докум.	Подп.	Дата		