Описание функциональных характеристик программного обеспечения

Data Quality Framework

Аннотация

Настоящий документ описывает назначение и функциональные характеристики программного обеспечения (далее – ПО) Data Quality Framework.

Исключительные права на ПО Data Quality Framework (далее – ПО DQF) принадлежат ООО «Клин Дейта».

Оглавление

Введение

Программное обеспечение Data Quality Framework (далее – ПО DQF) обеспечивает качественный контроль данных, которые рассматриваются как целостное образование, состоящее из взаимосвязанных частей. ПО DQF нацелено на мониторинг качества данных на различных уровнях и обнаружение ошибок по сформированным правилам и критериям проверок.

Функциональные возможности ПО DQF позволяют с помощью специальных правил и алгоритмов производить верификацию проверяемого объекта и его атрибутов и выявлять ошибки, а также настраивать проверки данных в соответствии с предметной областью и заданной конфигурацией (бизнес-логики) исполняемой проверки без доработки программного обеспечения.

1. Описание функциональных характеристик

ПО DQF обеспечивает:

- выполнение проверок данных по заданным алгоритмам и настраиваемым параметрам;
- выполнение проверок по массивам данных и единичных записей;
- выявление ошибок (противоречий) в имеющихся и вносимых данных;
- верификацию невалидных данных с возможностью из замены, в том числе в автоматизированном режиме;
- мониторинг и оценку качества данных на предмет полноты, достоверности и непротиворечивости.

Особенностью, реализованной ПО DQF, является оперирование высокоабстрагированными объектами, с определенным уровнем обобщения, при наличии метамодели данных. Такая отвязка от предметной области позволяет применять ПО DQF к различным областям и моделям описания данных. При этом ПО DQF обладает определенной автономностью в аспекте настройки любых логических проверок без привлечения разработчиков за счет возможности настройки проверок данных в соответствии с предметной областью с использованием конфигураций алгоритмов проверок.

ПО DQF может быть интегрировано в любую среду оркестрации, обеспечивает проверку объектов в памяти, без сохранения на диск.

При выполнении проверок ПО DQF осуществляет контроль за выполнением алгоритмов проверок, после чего предоставляет результаты проверок. При необходимости ПО DQF может осуществлять запросы дополнительных данных, необходимых для проведения проверок, в том числе из внешних информационных систем.

При росте нагрузки и количества запросов ПО DQF обладает качеством гибкого масштабирования. Также предусмотрена схема развертывания с горячим резервом с двумя активными экземплярами ПО и балансировщиком для обеспечения отказоустойчивости решения.

ПО DQF осуществляет автоматизированные проверки по трем направлениям:

- форматно-логический контроль:
  - наличие обязательных атрибутов;
  - соответствие атрибутов заданной длине и маске;

- соответствие атрибутов требуемым форматам – буквенно-числовые последовательности, непечатаемые символы, специальные символы;
- проверки внутри модели данных:
  - проверки по внутренним справочникам;
  - проверки на соответствие нормативно-правовым актам;
  - проверки на связность и непротиворечивость объектов.
- интеграционные проверки с использованием внешних источников:
  - проверки по внешним справочникам;
  - проверки по данным из Единой системы межведомственного электронного взаимодействия;
  - проверки по данным из Национальной системы управления данными;
  - проверки по данным из внешних информационных систем.

1.1. Алгоритмы проверок

Алгоритм проверки – это конечный набор шагов для обеспечения проверки данных посредством конечного количества операций.

В ПО DQF реализованы следующие типы алгоритмов:

- встроенные;
- базовые;
- комплексные.

Любой алгоритм имеет основную ветвь исполнения и может иметь несколько альтернативных ветвей исполнения (отмены, боковые ответвления и др.). Чтобы обеспечить возможность организации циклов и альтернативных веток исполнения, каждый алгоритм возвращает статус, логика определения которого задается отдельно для каждого алгоритма. В случае необходимости передачи данных между алгоритмами в процессе выполнения проверки, помимо статуса алгоритм может передать дополнительные данные через контекст исполнения.

Контекст исполнения состоит из следующих данных:

- входящий запрос:
  - исходный проверяемый объект;
  - переменные исполнения, заполненные на предыдущих шагах комплексного алгоритма.

- исходящий результат:
  - статус выполнения базового алгоритма;
  - переменные исполнения, при необходимости дополненные результирующими данными.

1.1.1. Встроенные алгоритмы проверок

Встроенные алгоритмы – это связующее звено, объединяющее базовые алгоритмы в единую последовательность исполнения – комплексный алгоритм.

ПО DQF содержит следующие встроенные алгоритмы:

- последовательное исполнение шагов – all-of (последовательности действий);
- альтернативная ветвь исполнения- otherwise (ветвления);
- итерирование по коллекции – for-each (циклы).

1.1.2. Базовые алгоритмы проверок

Базовый алгоритм проверки – это атомарная операция, производимая в определенном контексте над объектом/атрибутом объекта определенного типа. Из базовых алгоритмов составляются логически связанные цепочки, которые формируют сущность проверки. Базовые алгоритмы проверок реализованы в java-коде ПО DQF.

Примеры базовых алгоритмов проверок, реализованных в ПО DQF, представлены в таблице (Таблица 1).

Таблица 1 - Примеры базовых алгоритмов проверок, реализованных в ПО DQF

№	Содержание базового алгоритма проверки
1	Проверка существования объекта в системе Заказчика
2	Проверка связности данных в системе Заказчика (проверка ссылочной целостности)
3	Сравнение атрибута с константой
4	Проверка атрибута на нулевое значение
5	Проверка заполненности атрибута
6	Проверка атрибута на соответствие определённому типу
7	Проверка атрибута на соответствие регулярному выражению
8	Проверка атрибута путем математического преобразования по формуле
9	Проверка атрибута типа «дата» на соответствие временному интервалу (периоду)
10	Проверка атрибута на уникальность значения в пределах проверяемого объекта
11	Проверка атрибута на соответствие справочному значению внутренних

№	Содержание базового алгоритма проверки
	справочников
12	Проверка атрибута на соответствие значению, полученному из внутренней/внешней системы Заказчика
13	Суммирование неизвестного количества числовых атрибутов
14	Проверка ФИО на наличие опечатки в написании по справочнику
15	Проверка корректности идентификаторов юридического лица (основной государственный регистрационный номер, основной государственный регистрационный номер индивидуального предпринимателя, идентификационный номер налогоплательщика, код причины постановки на учет)
16	Проверка корректности идентификаторов физического лица (страховой номер индивидуального лицевого счета, идентификационный номер налогоплательщика)
17	Проверка корректности адреса, записанного одной строкой по справочнику (Федеральная информационная адресная система, Государственный адресный реестр)
18	Проверка корректности атрибутов гранулярного адреса по справочнику (Федеральная информационная адресная система, Государственный адресный реестр)
19	Интеграционный запрос связанных объектов
20	Сравнение атрибутов связанных объектов
21	Итерирование по связанным объектам
22	Проверка количества проверяемых объектов

1.1.3. Комплексные алгоритмы проверок

Комплексные алгоритмы проверок определяются в виде последовательностей шагов – вызовов базовых и встроенных алгоритмов.

1.1.4. Конфигурации алгоритмов проверок

Конфигурация алгоритма проверки содержит машиночитаемое описание последовательности применения параметризованных базовых алгоритмов с использованием встроенных алгоритмов. Конфигурация алгоритма проверки содержит следующие реквизиты:

- код порождаемой ошибки;
- список шагов алгоритма проверки, содержащих параметризованные конкретными значениями вызовы базовых алгоритмов, а также дополнительные параметры для формирования результата выполнения проверки.

Синтаксис конфигурации построен на основе языка YAML 1.2.2.

ПО DQF не осуществляет контроль версий конфигураций алгоритмов проверок.

1.2. Взаимодействие с брокером сообщений (API)

Программный интерфейс ПО DQF обеспечивает прием атомарных заданий на проверку и передачу результатов их выполнения в асинхронном режиме с использованием брокера сообщений. Атомарное задание на проверку – запрос на одно выполнение проверки данных по единичному алгоритму проверки.

В ПО DQF реализуется контракт взаимодействия с помощью сообщений в соответствии со стандартом AMQP (Advanced Message Queuing Protocol). Получение атомарных заданий на проверку реализуется из входящей очереди брокера сообщений. Передача результатов выполнения атомарных заданий на проверку реализуется в исходящую очередь брокера сообщений. Все сообщения имеют определенную структуру как способ запроса в брокер сообщений, состоящую из заголовочной части и тела сообщения.

Заголовочная часть сообщения-запроса и сообщения-ответа общая и содержит атрибуты, приведенные в таблице (Таблица 2).

Таблица 2 - Заголовочная часть сообщения-запроса и сообщения-ответа

№	Состав заголовка
1	Уникальный корреляционный идентификатор задания на проверку
2	Дата и время отправки сообщения

Тело сообщения-запроса на выполнение проверки представляет собой JSON-объект, который содержит атрибуты, приведенные в таблице (Таблица 3).

Таблица 3 - Тело сообщения-запроса

№	Атрибуты JSON-объекта	Примечание
1	Идентификатор настройки алгоритма проверки	
2	JSON-объект с данными проверяемого объекта	При передаче задания на выявление ошибки в рамках превентивного поиска по атрибутам или в границах объекта
3	Список JSON-объектов, содержащих данные проверяемых объектов одного типа	При передаче задания на выявление ошибки в рамках превентивного поиска по массиву объектов

4	Один идентификатор объекта внешней системы	При передаче задания на выявление ошибки в рамках инициативного поиска по атрибутам или в границах объекта
5	Список идентификаторов объектов одного типа	При передаче задания на выявление ошибки в рамках инициативного поиска по массиву объектов

Тело сообщения-ответа с результатами обработки сообщения-запроса на выполнение атомарной проверки по факту завершения выполнения алгоритма проверки представляет собой JSON-объект, который содержит атрибуты, приведенные в таблице (Таблица 4).

Таблица 4 - Тело сообщения-ответа

№	Атрибуты JSON-объекта	Примечание
1	Код результата обработки атомарного задания на проверку	
2	Опциональный список, содержащий пути в формате JSONPath к реквизитам объекта, в которых были обнаружены ошибки	В случае обнаружения ошибок
3	Опциональный список, содержащий значения дополнительных атрибутов проверяемого объекта	В случае обнаружения ошибок и необходимости формирования сообщений пользователям систем Заказчика, взаимодействующих с ПО DQF
4	Содержание системной ошибки	В случае возникновения системной ошибки

DQF не обеспечивает аутентификацию и авторизацию атомарных заданий на проверку из брокера сообщений.

С целью временного хранения полученных декларативных описаний (конфигураций) алгоритмов проверок применяется механизм кэширования.

1.3. Мониторинг и контроль работоспособности

ПО DQF осуществляет ведение журналов диагностических событий, в которых автоматически фиксируются возникающие нештатные ситуации и ошибки, и сохранение в этих журналах информации, необходимой для идентификации проблемы при возникновении аварийных ситуаций, либо ошибок в программном обеспечении. В таблице (Таблица 5) приведены примеры диагностических событий и характеристик.

Таблица 5 - Примеры диагностических событий и характеристик

№	Диагностическое событие/характеристика
1	Получение сообщения-запроса на атомарную проверку
2	Возврат сообщения-ответа, содержащего результат выполнения атомарной проверки
3	Запрос данных из внешней систем-источников
4	Получение данных из внешних систем-источников
5	Выполнение шагов алгоритма проверки
6	Время выполнения алгоритма проверки
7	Системные ошибки и предупреждения

Для контроля показателей работоспособности/сбоев ПО DQF при его функционировании с применением средств мониторинга программных компонентов, а также контроля корректности функционирования ПО DQF с применением сигнализаторов (извещателей), предусмотрены метрики, приведенные в таблице (Таблица 6).

Таблица 6 - Метрики работоспособности

№	Метрика работоспособности
1	Текущее состояние работоспособности сервиса
2	Количество одновременно исполняемых проверок
3	Количество закэшированных в данных момент конфигураций алгоритмов проверок
4	Количество закэшированных в данный момент объектов внешних систем Заказчика

2. Требования к программно-аппаратной платформе

2.1. Требования к аппаратной инфраструктуре

Процессор:

– Intel Xeon Ice Lake или новее (Intel Xeon Silver 4310 и более старшие версии) или AMD EPYC второго поколения или новее (AMD EPYC 7502P и более старшие версии);

– от 6 ядер.

ОЗУ:

– от 16 Гб.

2.2. Требования к программному обеспечению

Управление очередью сообщений:

– брокер сообщений (RabbitMQ, Kafka).

Сервер приложений:

– рекомендуемые операционные системы: «Альт 8 СП»;

– Open JDK 11 или Liberica JDK, с установленными актуальными обновлениями.

Веб сервер/балансировщик нагрузки:

– nginx версии 1.15.

ПО для организации отказоустойчивости сетевых сервисов и балансировки нагрузки:

– HAproxy версии 2.0.

Мониторинг:

– Prometheus версии 2.27.1;

– Grafana версии 7.5.7.

Логирование (ELK):

– Elasticsearch версии 7.11.2;

– Kibana версии 7.11;

– Fluent Bit версии 1.7.9.

2.3. Требования к сетевой инфраструктуре

Предъявляются следующие требования к сетевой инфраструктуре:

– пропускная способность канала между сервером приложений и внешними информационными системами, направляющими запросы в ПО DQF, составляет не менее 1 Гбит/с;

– пропускная способность каналов сервером приложений и брокером сообщений – не менее 1 Гбит/с.