Arenadata Catalog

Руководство Администратора ADC.DQF

Москва 2026

Arenadata Catalog

Оглавление

Arenadata Catalog

1 Введение

1.1 Полное наименование

Модуль Arenadata Catalog Data Quality Framework (ADC.DQF)

1.2 Область применения

Модуль ADC.DQF обеспечивает выполнение проверок данных по заданным алгоритмам и настраиваемым параметрам для массивов данных и единичных записей, выявление ошибок и противоречий в имеющихся и вносимых данных.

Модуль ADC.DQF предназначен для реализации следующих задач:

- создания/редактирования/просмотр правил для оценки качества данных;
- исполнения проверок качества данных по данным правилам;
- управления расписанием запуска проверок по правилам в рамках задач;
- запуска задач проверки качества данных по правилам;
- отслеживания статусов исполнения задач;
- отслеживания результатов задач;
- формирования отчётов по задачам в формате excel;
- создания/редактирования/просмотр пользовательских дашбордов (панелей с результатами статистики по задачам);
- предоставление результатов статистики по задачам в формате excel и на дашбордах в пользовательском интерфейсе;
- предоставление пользовательского интерфейса по всем вышеперечисленным пунктам;
- автоматическое формирование сообщений в интеграционную очередь kafka сигналов и результатов по задачам

1.3 Краткое описание возможностей

Модуль ADC.DQF:

- серверная часть реализована на базе java-технологий;
- клиентская часть реализована на базе JavaScript-технологий;
- предоставляет пользовательский интерфейс, отображаемый в современных браузерах;
- интегрируется с системами источниками при помощи JDBC, REST API и SOAP;
- интегрируется посредством API внутренних сервисов, с системами инициирующими проверки
- интегрируется с системами, потребляющими результаты задач проверки качества данных при помощи Kafka;
- настраивается посредством передачи конфигурационных параметров через файл конфигурации или переменные окружения

ADC.DQF позволяет обеспечить:

- соблюдение критериев качества данных;
- исполнение требований регуляторов;
- поддержку процессов цифровизации с применением методик Data Quality на новых платформах;
- соблюдение критериев качества данных при миграции и консолидации данных, предобработке данных;
- проактивное реагирование на инциденты качества и аномалии данных;

- централизацию проверок, реализацию методологий Data Quality и Data Governance в рамках одного интерфейса.

Модуль ADC.DQF состоит из структурных компонентов, приведённых в таблицах 1,2,3.

Таблица 1 Компоненты модуля ADC.DQF

Функциональный блок	Компоненты продукта
Сервис исполнения правил (СИП)	DQF Engine Standalone
Сервис расписаний задач (СРЗ)	DQF Scheduler
Сервис планирования задач (СПЗ)	DQF Planner
Сервис каталога правил (СКП)	DQF CAS
Сервис результатов «Агрегатор»	DQF Aggregator
Сервис дашбордов	DQF Dashboards Storage
Сервис отчетов	DQF Reporter
Веб-Портал	DQF Web App
Сервис авторизации	DQF Auth
Адаптер результатов для модуля сохранения	DQF Saver Adapter

Модули экосистемы Arenadata Catalog, которые дополняют ADC.DQF и обеспечивают дополнительную функциональность, описаны в таблице 2.

Таблица 2 Компоненты экосистемы ADC

Функциональный блок	Компоненты продукта
Модуль сохранения подробных результатов*	Result Saver
*обеспечивает функцию сохранения результатов в пользовательское s3 совместимое хранилище.	

Инфраструктурные компоненты, необходимые для эксплуатации ADC.DQF (Таблица 3)

Таблица 3 Дополнительные компоненты для среды модуля ADC.DQF

Функциональный блок	Компонент среды функционирования
Перенаправление логов	Vector - 0.28.0
Служебная база данных	Postgres 14 или любая совместимая с ней СУБД
Брокер сообщений	Kafka (предпочтительнее), RabbitMQ
Средство аутентификации	KeyCloak - 23.0
Хранение и поиск логов	Loki - 2.6.0, Grafana - 9.5.5
Сбор метрик	Prometheus - 2.41.0, Prometheus Graphite Exporter - 0.13.1
Единая точка входа	Nginx
Служба координации**	Zookeeper - 3.9
** используется для распространения и проверки лицензий	

1.4 Уровень подготовки администратора

Администратор ADC.DQF должен обладать высоким уровнем квалификации и практическим опытом выполнения работ по:

- установке и развёртыванию контейнерных серверных приложений;
- настройке переменных среды контейнеров и конфигурирования spring приложений;
- настройка логирования и мониторинга при помощи библиотеки log4j2.
- настройке программной части модулей, обладать знаниями и умением классифицировать и устранять возникающие ошибки.

1.5 Перечень эксплуатационной документации

Таблица 4 Перечень эксплуатационной документации

Наименование	Обозначение
Руководство пользователя	ADC.DQF РП
Руководство по языку Data Quality Language	ADC.DQF Руководство DQL
Инструкция по установке	ADC.DQF Инструкция по установке

2 Назначение и условия применения

Ниже представлена архитектурная схема ADC.DQF.

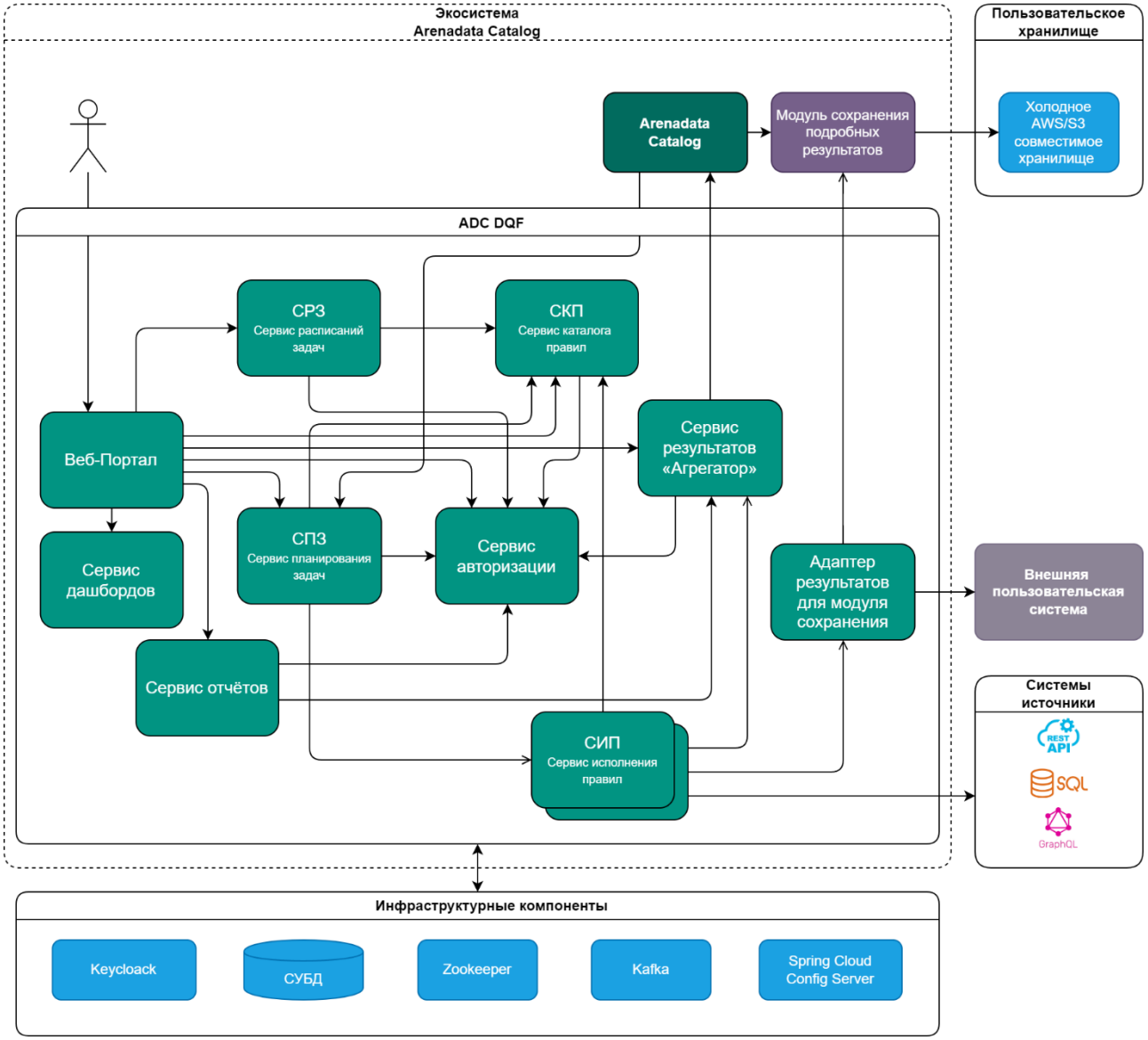


Рисунок 1. Архитектурная схема ADC.DQF.

2.1 Автоматизируемые виды деятельности и функции

Сервис каталога правил (СКП)

1) хранение и версионирование правил и групп правил;
2) предоставление информации по хранимым правилам и группам правил.

Сервис планирования задач (СПЗ):

1) Формирование задачи на проверку данных по перечню правил или группе правил, хранимых в СКП.
2) В рамках задачи отправка данных для проверки с указанием правила, по которому будет проводиться проверка в СИП;

Сервис расписаний задач (СРЗ):

1) Хранение данных о расписании запуска задач по группе
2) Формирование по расписанию задачи на проверку данных по группе в СПЗ;

Сервис результатов «Агрегатор»:

1) Сбор и хранение результатов проверок по задаче;
2) Идентификация принадлежности результатов запущенной задаче;
3) Идентификация принадлежности результатов задаче, запущенной по расписанию;
4) Идентификация завершения обработки результатов по задаче;
5) Информирование системы о завершении обработки результатов задачи;
6) Предоставление данных о результатах задач с возможностью фильтрации.

Сервис отчетов:

1) Получение данных от сервиса результатов по задаче
2) Формирование отчётов в формате excel

Сервис дашбордов:

1) хранение и предоставление данных о пользовательских дашбордах

Веб-портал:

1) отрисовка пользовательского интерфейса
2) предоставление пользователю возможностей использования вышеперечисленных сервисов в соответствии с предоставленными ему правами

Сервис авторизации:

1) чтение прав доступа из токена пользователя, либо интеграцией с Arenadata Catalog;
2) передача информации о доступных пользователю правах.

Сервис исполнения правил (СИП):

1) получение данных из СПЗ для проверки по указанному правилу;
2) получение правила из СКП;
3) исполнение правила над полученными данными в соответствии с возможностями языка (см. Руководство по языку DQL)
4) отправка результатов исполнения проверки

Адаптер результатов для модуля сохранения:

1) Чтение результатов проверки и событий сервисов с целью формирования сигналов для модуля сохранения
2) Формирование сообщений в интеграционную очередь kafka сигналов и результатов по задачам для модуля сохранения подробных результатов.

2.2 Условия, при соблюдении которых обеспечивается применение средства автоматизации в соответствии с назначением

2.2.1 Требования к вычислительной среде

Расчет требований к вычислительной среде для среды PROD готовится индивидуально в зависимости от многих факторов.

Минимальные требования к серверу приложения и компонентам продукта

- 16 ядер CPU x64, не менее 2 Ггц (рекомендуется платформа не старше Intel Cascade Lake);
- 16 Гб оперативной памяти;
- 14 Гб полезного дискового пространства HDD/SDD (с минимальной скоростью чтения 150МБ/сек);

Минимальные требования к серверу БД

- 8 ядер CPU x64, не менее 2 Ггц (рекомендуется платформа не старше Intel Cascade Lake);
- 16 Гб оперативной памяти;
- 50 Гб полезного дискового пространства SDD (с минимальной скоростью чтения 150МБ/сек);

2.2.2 Требования к среде исполнения ADC.DQF

1. Операционная система – Ubuntu 22.04.4 LTS, Ред ОС 7.3, ALT Server 10.1 (Mendelevium), Astra Linux Воронеж 1.7.
2. ПО Docker version 25.0.0, и выше, совместно с Docker compose version 1.29.2, и выше
3. Пользователь с правами sudo (или пользователь с правами использовать docker и запускать контейнеры)

3 Подготовка к работе

3.1 Состав и содержание дистрибутивного носителя данных

Архив с docker-образами ПО:

- dqf_images.zip

Архив с подготовленным каталогом стенда:

- dqf-demo-stand.zip

Файл лицензии:

- es.adc.license

Архив с докер образами содержит компоненты по перечню

Таблица 4. Компоненты модуля ADC.DQF

Функциональный блок	Компонент платформы
Сервис исполнения правил (СИП)	DQF Engine Standalone
Сервис расписаний задач (СРЗ)	DQF Scheduler
Сервис планирования задач (СПЗ)	DQF Planner
Сервис каталога правил (СКП)	DQF Complex Algorithms Storage Service
Сервис результатов	DQF Aggregator
Сервис дашбордов	DQF Dashboards Storage
Сервис отчетов	DQF Reporter
Веб-портал	DQF UI
Сервис авторизации	DQF Auth Service
Адаптер результатов для модуля сохранения	DQF Saver Adapter
Модуль сохранения подробных результатов*	Result Saver
Служебная база данных	Postgres 14 или совместимая
Брокер сообщений	Kafka
Средство аутентификации	KeyCloak - 23.0
Перенаправление логов	Vector - 0.28.0
Хранение и поиск логов	Loki - 2.6.0, Grafana - 9.5.5
Сбор метрик	Prometheus - 2.41.0,
	Prometheus Graphite Exporter - 0.13.1
Балансировщик нагрузки	Nginx
Служба координации	Zookeeper - 3.9

3.2 Порядок развёртывания и запуска модуля

Подробное описание порядка установки и запуска модуля описано в Инструкции по установке модуля ADC.DQF (раздел 1.5).

3.3 Порядок проверки работоспособности сервиса

Информацию о состоянии сервисов можно получить, выполнив команды ниже

Листинг 1 Получение информации о состоянии сервисов

```
docker ps
docker compose ps
docker compose logs <container name>
```

, а также при помощи http запроса по адресу: `{хост-сервиса:порт-сервиса}/actuator/health`

4 Описание операций

4.1 Обновление лицензии

Лицензия DQF подразумевает синхронизацию количества потоков между инстансами, которая происходит через zookeeper (/io/arenadc/dqf/es.adc.license/lease).

В случае, если файл недоступен, лицензия просрочена или невалидна - приложение отключит интеграционные интерфейсы до прочтения активного файла лицензии, health-состояние сервиса исполнения правил будет DOWN (/actuator/health или /actuator/health/licenseHealth).

Для установки/обновления лицензии необходимо получить файл лицензии у технической поддержки.

4.2 Конфигурирование ADC.DQF

Сервисы ADC.DQF настраиваются стандартным путём передачи конфигурационных параметров с помощью .env, yaml или properties.

Для всех сервисов модуля ADC.DQF в соответствии с docker compose определены специфичные конфигурационные файлы в соответствующих каталогах сервисов и общие для всех сервисов в каталоге spring-common:

- `application.yaml`;
- `application-dqf-kafka.yaml`;
- `application-keycloak.yaml`.

На примере логирования:

Уровень логирования, может быть, один из следующих:

• TRACE- самый детальный уровень логирования:
• DEBUG – логирование всех видов событий;
• INFO – логирование ошибок, предупреждений и сообщений;
• WARN – логирование ошибок и предупреждений;
• ERROR – логирование всех ошибок.

Настройка уровня логирования для всех сервисов модуля ADC.DQF выполняется в файле application.yaml в каталоге [spring-common]:

Листинг 2 Настройка уровня логирования для всех сервисов модуля ADC.DQF

```
logging:
  level:
    io:
      arenadc:
        dqf:
          service: info
```

Логи сохраняются в директории вида :

- docker/profile-stand/mnt/logs/<service>

например:

- docker/profile-stand/mnt/logs/aggregator
- docker/profile-stand/mnt/logs/auth

4.2.1 Конфигурационные параметры общие для всех backend сервисов

Общие конфигурационные параметры для сервисов перечислены ниже (Таблица 5), в примечании указывается обязательность параметра, значение по умолчанию и пример заполнения. В случае если в примечании не указано, что параметр обязательный, то параметр не является обязательным. Если не указано значение по умолчанию, то оно отсутствует.

Таблица 5 Общие параметры сервисов

Параметр	Описание	Примечание
`spring.config.import`	Дополнительные конфигурационные параметры.	**По умлочанию**: optional:configs-erver: **Пример**: configserver:http://localhost:8888
`spring.cloud.bootstrab.enabled`	Задействовать устаревший механизм bootstrap-а. Эквивалентно подключению зависимости spring-cloud-starter-bootstrap	**По умлочанию**: false: **Пример**: true
`spring.profiles.active`	Список профилей запуска приложения, разделенных запятыми	**Обязателен**. **По умолчанию:** keycloak **Пример:** **«**anonymous, dqf-kafka, gos-tech-audit»
`dqf.security.anonymous.principal`	Имя пользователя при анонимном доступе к АПИ. (профиль anonymous)	**Не применяется для СИП.** **По умолчанию**: anonymous-api
`spring.security.oauth2.*`	**Настройки OAuth2 аутентификация. Используются при профиле 'keycloak'.**	**Не применяется для СИП.**
`spring.security.oauth2.client.registration.keycloak.client-id`	Клиент backend-сервиса для подключения к Keycloak. Например: *dqf-planner*	**Обязательный**
`spring.security.oauth2.client.provider.keycloak.issuer-uri`	URL настроенного для приложения сервера аутентификации (IAM), например *keycloak*.	**Обязательный.** **Например**: http://localhost:8092/realms/DQF
`spring.security.oauth2.resourceserver.jwt.clock-skew`	Время, которое сервер будет считать переданный токен непросроченным (калибровка на случай рассинхронизации времени keycloak и application-сервера).	**По умолчанию:** PT60S
`spring.security.oauth2.resourceserver.jwt.validate-issuer`	Выполнять ли проверку поля iss JWT-токена.	**По умолчанию:** true
`spring.security.oauth2.client.provider.keycloak.user-name-attribute`	Название claim, содержащее имя пользователя.	**По умолчанию:** preferred_username
`spring.security.dqf.*`	**Параметры обращения к сервису аутентификации. Используются при профиле keycloack**	**Не применяются для СИП ,сервиса аутентификации, адаптера результатов и модуля сохранения подробных результатов**
`spring.security.dqf.auth-service-url`	URL настроенного DQF Auth Service для получения атрибутов доступа.	**Обязательный** **Пример**: http://localhost:8093/dqf-auth-service/.
`spring.security.dqf.cache.enabled`	Включение кэша атрибутов DQF.	**По умолчанию:** false
`spring.security.dqf.cache.ttl`	Время жизни кэша атрибутов DQF.	**PT5S**
`spring.security.dqf.cache.maxSize`	Максимальное количество токенов, для которых кэшируются атрибуты.	**15**
`gostech.audit.*`	**Конфигурационные параметры аудита, при профиле gostech-audit**	
`gostech.audit.enabled`	Создавать или нет бин с сервисом по отправке событий аудита. **Если значение false все остальные параметры не имеют значения**	**По умолчанию:** true
`gostech.audit.metamodel`	Ресурс в формате JSON с описанием метамодели. Мета-модель автоматически отправляется после старта приложения (пример см. ниже).	**По умолчанию:** classpath:gostech/metamodel.json

Параметр	Описание	Примечание
gostech.audit.send-metamodel-onstart	Отправлять метамодель на старте приложения	**По умолчанию:** true
gostech.audit.url	**Хост, на котором находится сервис аудита**	**Обязательный**
gostech.audit.connect-timeout-millis	Таймаут в миллисекундах на подключение к сервису аудита(>0)	**По умолчанию:** 10000
gostech.audit.read-timeout-millis	Таймаут в миллисекундах на ожидание ответа от сервиса аутентификации(>0)	**По умолчанию:** 10000
gostech.audit.tls.key-store.store-type	Тип хранилища приватных ключей для аутентификации на сервере по ЭЦП (PKCS12, JKS, ...)	
gostech.audit.tls.key-store.store-file	Ресурс, из которого загружается хранилище (файл с расширением .p12, .jks, ...)	
gostech.audit.tls.key-store.store-password	Ресурс, в котором записан пароль для доступа к хранилищу (текстовый файл с паролем в первой строке)	
gostech.audit.tls.trust-store.store-type	Тип хранилища с сертификатом сервера (публичный ключ + информация о владельце) для аутентификации сервера и установки защищенного канала (HTTPS/TLS)	
gostech.audit.tls.trust-store.store-file	Ресурс, из которого загружается сертификат/цепочка сертификатов (файл с расширением .p12, .jks, ...)	
gostech.audit.tls.trust-store.store-password	Ресурс, в котором записан пароль для доступа к хранилищу (текстовый файл с паролем в первой строке)	
dqf.audit.web.ignore.ant-matchers	Набор (массив строк) ant-матчеров для ресурсов, доступ к которым не будет генерировать события аудита.	**По умолчанию:** ['/actuator/prometheus']
spring.kafka.*	Общая конфигурация соединения Kafka.	**Не применяется для сервисов:** авторизации, отчётов, дашбордов, модуля подробных результатов
spring.rabbitmq.*	Общая конфигурация соединения RabbitMQ.	**Не применяется для сервисов:** авторизации, отчётов, дашбордов, адаптера к модулю подробных результатов и модуля подробных результатов.
spring.datasource.*	**Конфигурация бд сервиса**	**Не применяется для сервисов:** исполнения проверок, авторизации, адаптера к модулю подробных результатов и модуля подробных результатов.
spring.datasource.url	Адрес базы данных	**Пример: jdbc:postgresql://postgres-cas:5432/cas**
spring.datasource.username	Имя пользователя для подключения к БД	**cas-a**
spring.datasource.password	Пароль для подключения к БД	**cas-p**
dqf.events.*	Настройка отправки событий	**Применяется для сервисов:** расписаний, результатов, каталога правил и планировщика.
dqf.events.enabled	Включение событий микросервисов, обеспечивающих связанную работу DQF.	**По умолчанию:** true
dqf.events.queue	Очередь, в которую фоновый процесс будет отправлять новые события	**Пример:** event_q
oauth.*	Параметры аутенификации сервисов	**Применяется для сервисов: расписаний, отчётов, планировщка.**

Параметр	Описание	Примечание
`oauth.basic.enabled`	Включение/отключение получения токена аутентификации в сервисе аутентификации по контракту GET basic. **Если значение false все остальные параметры *.basic.* не имеют значения**	**По умолчанию:** false
`oauth.basic.username`	Имя пользователя для Basic аутентификации	**По умолчанию:** null
`oauth.basic.password`	Пароль для Basic аутентификации	**По умолчанию:** null
`oauth.keycloak.enabled`	Включение/отключение получения токена аутентификации в сервисе аутентификации Keycloak. **Если значение false все остальные параметры *.keycloak.* не имеют значения**	**По умолчанию:** false
`oauth.keycloak.username`	Имя пользователя для Keycloak аутентификации	**Обязательный** **По умолчанию:** null
`oauth.keycloak.password`	Пароль для Keycloak аутентификации	**Обязательный** **По умолчанию:** null
`oauth.keycloak.clientId`	Идентификатор клиента для Keycloak аутентификации	**Обязательный** **По умолчанию:** null
`oauth.keycloak.clientSecret`	Секрет клиента для Keycloak аутентификации	**Обязательный** **По умолчанию:** null
`oauth.*.url`	Хост, на котором находится сервис аутентификации	**Обязательный** **По умолчанию:** null
`oauth.*.connectTimeoutMillis`	Таймаут в миллисекундах на подключение к сервису аутентификации(>0). Значение по-умолчанию зависит от версии reactor.netty	**По умолчанию:** 30000
`oauth.*.readTimeoutMillis`	Таймаут в миллисекундах на ожидание ответа от сервиса аутентификации(>0). Значение по-умолчанию зависит от версии reactor.netty	**По умолчанию:** 30000
`oauth.*.tls.key-store.store-type`	Тип хранилища приватных ключей для аутентификации на сервере по ЭЦП (PKCS12, JKS, ...)	
`oauth.*.tls.key-store.store-file`	Ресурс, из которого загружается хранилище (файл с расширением .p12, .jks, ...)	
`oauth.*.tls.key-store.store-password`	Ресурс, в котором записан пароль для доступа к хранилищу (текстовый файл с паролем в первой строке)	
`oauth.*.tls.key-store.key-password`	Ресурс, в котором записан пароль для доступа к ключу (текстовый файл с паролем в первой строке). Если не задан, используется ключ от хранилища	
`oauth.*.tls.trust-store.store-type`	Тип хранилища с сертификатом сервера (публичный ключ + информация о владельце) для аутентификации сервера и установки защищенного канала (HTTPS/TLS)	
`oauth.*.tls.trust-store.store-file`	Ресурс, из которого загружается сертификат/цепочка сертификатов (файл с расширением .p12, .jks, ...)	
`oauth.*.tls.trust-store.store-password`	Ресурс, в котором записан пароль для доступа к хранилищу (текстовый файл с паролем в первой строке)	
`oauth.*.tls.verify-host-name`	Проверять ли серверный сертификат на соответствие имени хоста или IP-адреса	

4.2.1 Конфигурационные .env-параметры для Веб-портала

Таблица 6 Параметры настройки Web-App

Параметр	Описание	Пример заполнения
USE_AUTH_SERVICE	Включает или выключает использование внешнего auth-сервиса. Если значение true, приложение пытается получать auth-конфигурацию через сервис. Если false, используется anonymous-режим.	false
AUTH_SERVICE_API_ORIGIN	Базовый origin для auth-сервиса. Используется как префикс, к которому добавляется PUBLIC_AUTH_SERVICE_API_URI.	“http://localhost:8079”
PUBLIC_AUTH_SERVICE_API_URI	Публичный URI сервиса авторизации API	“/auth-api”
DQF_WEB_BASE_URL	Базовый URL веб-приложения	“/dqf”
PUBLIC_COMPLEX_ALGORITHMS_STORAGE_API_URL	URL для API каталога проверок	"/cas-api"
PUBLIC_PLANNER_API_URL	URL для API планировщика задач	"/planner-api"
PUBLIC_SCHEDULER_API_URL	URL для API сервиса расписаний	"/scheduler-api"
PUBLIC_AGGREGATOR_API_URL	URL для API сервиса результатов	"/aggregator-api"
PUBLIC_DASHBOARDS_STORAGE_API_URL	URL для API сервиса дашбордов	"/dashboards-storage-api"
PUBLIC_REPORTER_API_URL	URL для API сервиса отчётов	"/reporter-api"
PUBLIC_ENGINE_PROBE_API_URL	URL для API пробного запуска СИП	"/engine-probe-api"
EXPLORER_ADC_BASE_URI	Базовый URI для API ADC.	"/adc-api"

4.2.2 Конфигурационные параметры сервиса исполнения правил (СИП) и сервиса проверки правил (СПП)

Общие параметры СИП и СПП (раздел 4.2.1) требуют настройки профиля, очередей, и авторизации при необходимости, так же может получать параметры от сервера конфигураций. СИП от СПП отличается тем, что `spring.profiles.active` содержит значение `probe`. Ниже приводятся параметры специфичные для СИП и СПП.

Ниже (Таблица 7, Таблица 8, Таблица 9) описаны параметры необходимые для обработки сообщений, параметров кэширования и интеграции с каталогом СКП). Параметры

Таблица 7 Параметры СИП в части обработки сообщений.

Параметр	Что делает	Примечание
dqf.controller.type	Тип коннектора к внешней очереди. Поддерживается 3 взаимоисключающих значения: KAFKA, RABBIT, PROBE.	**Обязателен**. **Пример**: KAFKA **для СПП**: PROBE
dqf.controller.queue.read.name	Очередь запросов на валидацию	**Обязателен**. **Пример**: read_q

Параметр	Что делает	Примечание
dqf.controller.queue.write.name	Очередь ответов с результатами валидации. Не применяется для СПП	**Обязателен**. **Пример**: write_q
dqf.controller.queue.write.success.name	Очередь для успешных результатов (status=true). Не применяется для СПП	**По умолчанию**: dqf.controller.queue.write.name. **Пример**: write_success_q
dqf.controller.queue.write.failure.name	Очередь для неуспешных результатов (status=false). Не применяется для СПП	**По умолчанию**: dqf.controller.queue.write.name. **Пример**: write_failure_q
dqf.controller.queue.write.error.name	Очередь для ошибок исполнения. По умолчанию:	**По умолчанию**: dqf.controller.queue.write.name. **Пример**: dlx_q

Таблица 8 Параметры кэширования

Параметр	Что делает	Примечание
dqf.cache.noop-on-error	Игнорирование ошибок при работе с кэшем. Операция будет продолжена без кэша.	**По умолчанию**:true

Таблица 9 Параметры интеграции с сервисом каталога правил – Complex Algorithm Storage(CAS)

Параметр	Что делает	Примечание
dqf.complex-algorithm.storage.cas-service.* - и		
dqf.complex-algorithm.storage.cas-service.enabled	Включение чтения конфигураций правил из сервиса CAS	По умолчанию: false
dqf.complex-algorithm.storage.cas-service.url	Базовая ссылка на сервис CAS	Обязательно, при включении интеграции. Пример: http://cas.interanal.arenadc.io
dqf.complex-algorithm.storage.cas-service.connectTimeoutMillis	Таймаут на подключение к внешнему сервису в миллисекундах.	По умолчанию 1000
dqf.complex-algorithm.storage.cas-service.readTimeoutMillis	Таймаут ожидания ответа от внешнего сервиса в миллисекундах.	По умолчанию 1000
dqf.complex-algorithm.storage.cas-service.tls.*	Настройки TLS.	Не обязательно, при незащищенном соединении или использовании настроек JRE.
dqf.complex-algorithm.storage.cas-service.auth.enabled	Флаг включения авторизации в CAS. Авторизация будет работать при получении статуса 401	По умолчанию: false
dqf.complex-algorithm.storage.cas-service.auth.url	Ссылка получения авторизационного токена для CAS.	Да Пример: http://keycloak.interanal.arenadc.io/realms/SomeRealm/protocol/openid-connect/token
dqf.complex-algorithm.storage.cas-service.auth.username	Имя пользователя для Basic аутентификации.	Обязателен при включении авторизации. Пример: user
dqf.complex-algorithm.storage.cas-service.auth.password	Пароль для аутентификации.	Обязателен при включении авторизации. Пример: pwd
dqf.complex-algorithm.storage.cas-service.auth.clientId	Идентификатор клиента для аутентификации.	Обязателен при включении авторизации. Пример: dqf-standalone
dqf.complex-algorithm.storage.cas-service.auth.grantType	Тип гранта.	По умолчианию password

Параметр	Что делает	Примечание
`dqf.complex-algorithm.storage.cas-service.auth.connectTimeoutMillis`	Таймаут в миллисекундах на подключение к сервису аутентификации.	`1000`
`dqf.complex-algorithm.storage.cas-service.auth.readTimeoutMillis`	Таймаут в миллисекундах на ожидание ответа от сервиса аутентификации.	`1000`
`dqf.complex-algorithm.storage.cas-service.auth.tls.*`	Настройки TLS.	Не обязательно, при незащищенном соединении или использовании настроек JRE.
`dqf.complex-algorithm.storage.cas-service.cache.enabled`	Включение кэширования при получении списка всех доступных алгоритмов из сервиса CAS	По умолчанию: `true`
`dqf.complex-algorithm.storage.cas-service.cache.all-ids-expire-after-write`	Время жизни списка всех доступных алгоритмов в кэше в формате `java.time.Duration`	Да, при включении кеширования. По умолчанию: PT1S
`dqf.complex-algorithm.storage.cas-service.page-size`	Количество алгоритмов, возвращаемых в одном запросе при получении списка. По умолчанию: 500. ВАЖНО: Увеличение может потребовать соответствующего увеличения буфера для десериализации ответа через настройку `spring.codec.max-in-memory-size`	Нет

Для корректировки значения spring.codec.max-in-memory-size можно воспользоваться формулой:

$$(pagesize * \text{recordSizeInBytes}) + \text{pageResponseOverhead}$$

, где:

- recordSizeInBytes - размер сериализованного вида одной записи об алгоритме = 1000 байт (algorithmId максимальной длины на кириллице, форматирование - pretty JSON)
- pageResponseOverhead - накладные расходы на информацию о странице = 200 байт. Например: dqf.complex-algorithm.storage.cas-service.page-size = 5000 соответствует spring.codec.max-in-memory-size = 5MB.

Свойства TLS для CAS вида dqf.complex-algorithm.storage.cas-service.tls.* описаны без префикса в формате `*.tls.*` (Таблица 10)

Таблица 10. Типовые настройки TLS

Параметр	Что делает	Обязательно	Пример
`*.tls.verifyHostName`	Проверять ли серверный сертификат на соответствие имени хоста или IP-адреса	Нет.	`true`
`*.tls.keyStore.*`	Настройки хранилища приватных ключей.	Нет.	
`*.tls.keyStore.storeType`	Тип KeyStore.	Нет.	`SunPKCS11`
`*.tls.keyStore.storeFile`	Ресурс, в котором хранится KeyStore.	Да.	`classpath:keystore/path`
`*.tls.keyStore.storePassword`	Ресурс с паролем для доступа к хранилищу KeyStore.	Да.	`file:/keystore/password/path`
`*.tls.keyStore.storePassword`	Ресурс с паролем для доступа к хранилищу KeyStore.	Да.	`file:/keystore/password/path`

`*.tls.trustStore.*`	Настройки хранилища доверенных сертификатов.	Нет.	
`*.tls.trustStore.storeType`	Тип TrustStore.	Нет.	`SunPKCS11`
`*.tls.trustStore.storePassword`	Ресурс с паролем для доступа к хранилищу KeyStore.	Да.	`file:/keystore/password/path`
`*.tls.trustStore.storePassword`	Ресурс с паролем для доступа к хранилищу KeyStore.	Да.	`file:/keystore/password/path`

Доступна конфигурация для записи log-файлов в формате `json` в формате `elastic`. При включении логи будут писаться в файл `spring.log`.

Таблица 11. Настройка записи лог-файлов в yaml.

Параметр	Что делает	Обязательно	Пример
`logging.threshold.file`	Настройка уровня логирования в файл	Нет, по умолчанию: OFF - логирование в файл выключено	`INFO`

Обратить внимание:

- У приложения должны быть права на директорию записи логов.
- Конфигурация происходит исключительно env-переменными.

Таблица 12. Конфигурация логов env-переменными.

Параметр	Что делает	Обязательно	Пример
`LOG_TEMP`	Путь до **директории** логов	Нет	`/app/log`
`LOG_PATH`	Путь до **директории** логов. Приоритетнее LOG_TEMP	Нет	`/app/log`
`LOG_FILE`	Полный путь до log-**файла** с названием. Приоритетнее LOG_PATH	Нет	`/app/log/elk-formatted.log`
`LOGBACK_ROLLINGPOLICY_FILE_NAME_PATTERN`	Паттерн названия файла для ролловер-файлов	Нет	`/app/log/archive/old-log.%i.gz`
`LOGBACK_ROLLINGPOLICY_CLEAN_HISTORY_ON_START`	Очищать ли логи при рестарте. По-умолчанию false	Нет	`true`
`LOGBACK_ROLLINGPOLICY_MAX_FILE_SIZE`	Максимальный размер файла лога. По-умолчанию 10MB	Нет	`100MB`
`LOGBACK_ROLLINGPOLICY_TOTAL_SIZE_CAP`	Максимальный размер файлов логов. По-умолчанию не ограничен: 0	Нет	`200MB`

LOGBACK_ROLLINGPOLICY_MAX_HISTORY	Максимальное количество исторических файлов логов. По-уполчанию 7	Нет	10

4.2.2.1 Конфигурация хранилища отложенных операций СИП

Некоторые алгоритмы запускают отложенную операцию, храня состояние для восстановления в хранилище отложенного состояния.

Приложение поддерживает несколько реализаций хранилища отложенного состояния:

- in-memory хранилище отложенных операций (execution-data-store-in-mem) – хранит контекст исполнения в памяти приложения
- redis хранилище отложенных операций (execution-data-store-redis) – хранит контекст исполнения во внешнем redis хранилище

Одновременно может быть включено только одно хранилище.

Для корректной работы redis-хранилища - версия redis должна быть не ниже 6.2.3

Таблица 13. Настройки хранилища отложенных состояний

Параметр	Что делает	Обязательно	Значение по умолчанию
engine.data-store.mem.enabled	Включение in-memory хранилища	Нет	false
engine.data-store.redis.enabled	Включение redis-хранилища	Нет	false
spring.data.redis.*	Конфигурация redis-соединения по умолчанию	Условно	-

4.2.2.2 Конфигурация http-модуля СИП

Модуль алгоритмов СИП, который предоставляет общие механизмы взаимодействия HTTP, в частности настройку клиентов http. Ниже приводится список конфигурационных параметров.

Таблица 14 Конфигурация http-модуля СИП

Конфигурационный параметр	Описание
engine.algo.http-request.connect-timeout-millis	Таймаут на подключение к внешнему сервису в миллисекундах. Будет использован, если в конфигурации HttpRequest не указан client.
engine.algo.http-request.read-timeout-millis	Таймаут ожидания ответа от внешнего сервиса в миллисекундах. Будет использован, если в конфигурации HttpRequest не указан client.
engine.algo.http-request.tls.*	Настройки TLS. Будут использованы, если в конфигурации HttpRequest не указан client.
engine.algo.http-request.tls.key-store.store-type	Тип хранилища приватных ключей для аутентификации на сервере по ЭЦП (PKCS12, JKS, ...)
engine.algo.http-request.tls.key-store.store-file	Ресурс, из которого загружается хранилище (файл с расширением .p12, .jks, ...)
engine.algo.http-request.tls.key-store.store-password	Ресурс, в котором записан пароль для доступа к хранилищу (текстовый файл в кодировке UTF-8 с паролем в первой строке)
engine.algo.http-request.tls.key-store.key-password	Ресурс, в котором записан пароль для доступа к ключу (текстовый файл в кодировке UTF-8 с паролем в первой строке). Если не задан, используется ключ от хранилища

`engine.algo.http-request.tls.trust-store.store-type`	Тип хранилища с сертификатом сервера (публичный ключ + информация о владельце) для аутентификации сервера и установки защищенного канала (HTTPS/TLS)
`engine.algo.http-request.tls.trust-store.store-file`	Ресурс, из которого загружается сертификат/цепочка сертификатов (файл с расширением .p12, .jks, ...)
`engine.algo.http-request.tls.trust-store.store-password`	Ресурс, в котором записан пароль для доступа к хранилищу (текстовый файл в кодировке UTF-8 с паролем в первой строке)
`engine.algo.http-request.tls.verify-host-name`	Проверять ли серверный сертификат на соответствие имени хоста или IP-адреса
`engine.algo.http-request.caching.*`	Конфигурация кэша.
`engine.algo.http-request.caching.caffeine-managers[`*CM*`]`	*CM*=0, 1... Конфигурация менеджеров кэша.
`engine.algo.http-request.caching.caffeine-managers[`*CM*`].name`	Имя, с которым будет зарегистрирован бин `org.springframework.cache.caffeine.CaffeineCacheManager` в Spring-контексте и по которому к нему можно обратиться в параметре `engine.algo.http-request.caching.wrappers[`*W*`]`.cache-manager. Не может быть пустым.
`engine.algo.http-request.caching.caffeine-managers[`*CM*`].maximum-size`	Максимальное количество запросов, которые могут быть закэшированы (>= 0)
`engine.algo.http-request.caching.caffeine-managers[`*CM*`].expire-after-write`	Время жизни записи в кэше. Пример: `PT1S`. Должно быть положительным.
`engine.algo.http-request.caching.caffeine-managers[`*CM*`].use-weak-references`	Использовать ли для значения в кэше слабые ссылки. По умолчанию false. true применяется бля борьбы с утечкой памяти. Включайте при риске утечек памяти, если данные маленькие и предсказуемые, можно не применять.
`engine.algo.http-request.caching.wrappers[`*W*`]`	*W*=0, 1... Конфигурация кэширующих оберток. Добавляют кэширование запросов.
`engine.algo.http-request.caching.wrappers[`*W*`].name`	Имя, с которым будет зарегистрирован бин кэширующей обертки в Spring-контексте и по которому к нему можно обратиться в параметре `engine.algo.http-request.clients[`*CL*`].wrappers-refs[WR]`. Не может быть пустым.
`engine.algo.http-request.caching.wrappers[`*W*`].cache-manager`	Идентификатор бина `org.springframework.cache.CacheManager`. Если не задан, будет использован primary бин подходящего типа. Если более 1-го кандидата, будет выброшено стандартное исключение `NoUniqueBeanDefinitionException`
`engine.algo.http-request.caching.wrappers[`*W*`].cache-name`	Имя кэша. Не может быть пустым.
`engine.algo.http-request.clients[`*CL*`]`	*CL*=0, 1... Конфигурация клиентов. Определяют специфическую настройку для клиентов: base URL, фильтры запроса, параметры соединения.
`engine.algo.http-request.clients[`*CL*`].name`	Имя, с которым будет зарегистрирован бин клиента в Spring-контексте и по котору к нему можно обратиться из алгоритма. Не может быть пустым.
`engine.algo.http-request.clients[`*CL*`].base-url`	Базовый URL. Путь запроса, либо часть пути запроса. Будет использован, если в конфигурации HttpRequest указан client.
`engine.algo.http-request.clients[`*CL*`].connect-timeout-millis`	Таймаут на подключение к внешнему сервису в миллисекундах. Будет использован, если в конфигурации HttpRequest указан client.
`engine.algo.http-request.clients[`*CL*`].read-timeout-millis`	Таймаут ожидания ответа от внешнего сервиса в миллисекундах. Будет использован, если в конфигурации HttpRequest указан client.
`engine.algo.http-request.clients[`*CL*`].tls.*`	Настройки TLS. Будут использованы, если в конфигурации HttpRequest указан client.
`engine.algo.http-request.clients[`*CL*`].tls.key-store.store-type`	Тип хранилища приватных ключей для аутентификации на сервере по ЭЦП (PKCS12, JKS, ...)
`engine.algo.http-request.clients[`*CL*`].tls.key-store.store-file`	Ресурс, из которого загружается хранилище (файл с расширением .p12, .jks, ...)

`engine.algo.http-request.clients[`*CL*`].tls.key-store.store-password`	Ресурс, в котором записан пароль для доступа к хранилищу (текстовый файл в кодировке UTF-8 с паролем в первой строке)
`engine.algo.http-request.clients[`*CL*`].tls.key-store.key-password`	Ресурс, в котором записан пароль для доступа к ключу (текстовый файл в кодировке UTF-8 с паролем в первой строке). Если не задан, используется ключ от хранилища
`engine.algo.http-request.clients[`*CL*`].tls.trust-store.store-type`	Тип хранилища с сертификатом сервера (публичный ключ + информация о владельце) для аутентификации сервера и установки защищенного канала (HTTPS/TLS)
`engine.algo.http-request.clients[`*CL*`].tls.trust-store.store-file`	Ресурс, из которого загружается сертификат/цепочка сертификатов (файл с расширением .p12, .jks, ...)
`engine.algo.http-request.clients[`*CL*`].tls.trust-store.store-password`	Ресурс, в котором записан пароль для доступа к хранилищу (текстовый файл в кодировке UTF-8 с паролем в первой строке)
`engine.algo.http-request.clients[`*CL*`].tls.verify-host-name`	Проверять ли серверный сертификат на соответствие имени хоста или IP-адреса
`engine.algo.http-request.clients[`*CL*`].wrappers-refs[`*WR*`]`	*WR*=0, 1... идентификаторы бинов-оберток, вызываемых при отправке запроса. Например, можно использовать для кэширования.
`engine.algo.http-request.clients[`*CL*`].filters-refs[`*FR*`]`	*FR*=0, 1... идентификаторы бинов-фильтров, вызываемых при отправке запроса. Например, можно использовать для аутентификации.
`engine.algo.http-request.async.*`	Свойства асинхронного выполнения алгоритма HttpRequest.
`engine.algo.http-request.async.enabled`	Флаг включения асинхронного выполнения алгоритма HttpRequest. По-умолчанию `false`.
`engine.algo.http-request.async.parallelism`	Максимальное количество одновременно выполняющихся асинхронных запросов HttpRequest. Должно быть заполнено, если асинхронное выполнение включено. Должно быть положительным (`>0`).
`engine.algo.http-request.async.queue-size`	Максимальный размер очереди ожидающих выполнения асинхронных запросов. При достижении лимита - запросы будут совершаться в синхронном режиме. Должно быть заполнено, если асинхронное выполнение включено. Должно быть не отрицательным (`>=0`).

Листинг 3 Пример конфигурации клиентов http

```
engine:
  algo:
    http-request:
      connect-timeout-millis: 3000
      read-timeout-millis: 3000
      caching:
        caffeine-managers:
          - name: caffeineCacheManagerHttpRequest
            maximum-size: 1000
            expire-after-write: PT10M
            use-weak-references: false
        wrappers:
          - name: cachingWrapperforTest1
            cache-manager: caffeineCacheManagerHttpRequest
            cache-name: cache-for-graphql
      clients:
        - name: testClient
          base-url: 'http://localhost:8080/'
          connect-timeout-millis: 2000
          read-timeout-millis: 3000
          wrappers-refs:
            - cachingWrapperForTest1
          filters-refs:
            - basicAuthFilter
        - name: testClient2
          base-url: 'http://localhost:8081'
          connect-timeout-millis: 4000
          read-timeout-millis: 6000
          wrappers-refs:
            - redisCachingWrapper
          filters-refs:
            - oauth2AuthFilter
      async:
```

```
        # Алгоритм HttpRequest конфигурируется в асинхронном режиме
        enabled: true
        # Максимальное количество одновременно выполняемых асинхрон-
ных запросов
        parallelism: 50
        # Размер очереди асинхронных запросов
        # если одновременно придет больше чем parallelism+queueSize
- они будут выполняться синхронно
        queueSize: 250
```

4.2.2.3 Конфигурация graphQL-модуля СИП

GraphQL-модуль предоставляет базовый алгоритм для выполнения GraphQL-запросов и HTTP-клиент для обращения к внешнему GraphQL-эндпоинту. Настройки модуля сосредоточены под префиксом `graphql.client.*` и применяются глобально для всего модуля (то есть конфигурируется один GraphQL-endpoint на экземпляр приложения).

При необходимости обращения к нескольким GraphQL-сервисам в рамках одного экземпляра приложения рекомендуется использовать HTTP-модуль (HttpRequest)

Настройки вида `engine.algo.jdbc.*` содержит список клиентов jdbc, для использования в алгоритме jdbc.

Таблица 15. Настройки клиентов jdbc.

Параметр	Что делает	Обязательно	Значение по умолчанию
graphql.client.enabled	Включение/отключение graphql клиента	Нет	true
graphql.client.url	Хост, на котором находится graphql сервис	Да	null
graphql.client.connectTimeoutMillis	Таймаут в миллисекундах на подключение к graphql-сервису(>0). Значение по-умолчанию зависит от версии reactor.netty	Нет	30000
graphql.client.readTimeoutMillis	Таймаут в миллисекундах на ожидание ответа от graphql-сервиса(>0). Значение по-умолчанию зависит от версии reactor.netty	Нет	30000
graphql.client.tls.key-store.store-type	Тип хранилища приватных ключей для аутентификации на сервере по ЭЦП (PKCS12, JKS, ...)	Нет	
graphql.client.tls.key-store.store-file	Ресурс, из которого загружается хранилище (файл с расширением .p12, .jks, ...)	Нет	
graphql.client.tls.key-store.store-password	Ресурс, в котором записан пароль для доступа к хранилищу (текстовый файл с паролем в первой строке)	Нет	
graphql.client.tls.key-store.key-password	Ресурс, в котором записан пароль для доступа к ключу (текстовый файл с паролем в первой строке). Если не задан, используется ключ от хранилища	Нет	
graphql.client.tls.trust-store.store-type	Тип хранилища с сертификатом сервера (публичный ключ + информация о владельце) для аутентификации сервера и установки защищенного канала (HTTPS/TLS)	Нет	
graphql.client.tls.trust-store.store-file	Ресурс, из которого загружается сертификат/цепочка сертификатов (файл с расширением .p12, .jks, ...)	Нет	

graphql.client.tls.trust-store.store-password	Ресурс, в котором записан пароль для доступа к хранилищу (текстовый файл с паролем в первой строке)	Нет	
graphql.client.tls.verify-host-name	Проверять ли серверный сертификат на соответствие имени хоста или IP-адреса	Нет	
graphql.client.cache.enabled	Включение/отключение кэширования ответа от graphql сервиса	Нет	false
graphql.client.cache.maximumSize	Максимальный размер хранения в используемой реализации. Значение по-умолчанию подразумевает что размер кэша не ограничен	Нет	-1
graphql.client.cache.expireAfterWrite	Время жизни записи в кэше. Измеряется в секундах. Значение по-умолчанию подразумевает что время жизни не ограничено	Нет	-1
graphql.client.cache.useWeakReference	Использование weakReference в кэше	Нет	false
graphql.client.oauth.basic.enabled	Включение/отключение получения токена аутентификации в сервисе аутентификации. Эта группа приоритетнее при формировании клиента, чем глобальная oauth.basic.*. **Если значение false все остальные параметры не имеют значения**	Нет	false
graphql.client.oauth.basic.username	Имя пользователя для Basic аутентификации	Да	null
graphql.client.oauth.basic.password	Пароль для Basic аутентификации	Да	null
graphql.client.oauth.basic.url	Хост, на котором находится сервис аутентификации	Да	null
graphql.client.oauth.basic.connectTimeoutMillis	Таймаут в миллисекундах на подключение к сервису аутентификации(>0). Значение по-умолчанию зависит от версии reactor.netty	Нет	30000
graphql.client.oauth.basic.readTimeoutMillis	Таймаут в миллисекундах на ожидание ответа от сервиса аутентификации(>0). Значение по-умолчанию зависит от версии reactor.netty	Нет	30000
graphql.client.oauth.basic.tls.key-store.store-type	Тип хранилища приватных ключей для аутентификации на сервере по ЭЦП (PKCS12, JKS, ...)	Нет	
graphql.client.oauth.basic.tls.key-store.store-file	Ресурс, из которого загружается хранилище (файл с расширением .p12, .jks, ...)	Нет	
graphql.client.oauth.basic.tls.key-store.store-password	Ресурс, в котором записан пароль для доступа к хранилищу (текстовый файл с паролем в первой строке)	Нет	
graphql.client.oauth.basic.tls.key-store.key-password	Ресурс, в котором записан пароль для доступа к ключу (текстовый файл с паролем в первой строке). Если не задан, используется ключ от хранилища	Нет	
graphql.client.oauth.basic.tls.trust-store.store-type	Тип хранилища с сертификатом сервера (публичный ключ + информация о владельце) для аутентификации сервера и установки защищенного канала (HTTPS/TLS)	Нет	
graphql.client.oauth.basic.tls.trust-store.store-file	Ресурс, из которого загружается сертификат/цепочка сертификатов (файл с расширением .p12, .jks, ...)	Нет	

graphql.client.oauth.basic.tls.trust-store.store-password	Ресурс, в котором записан пароль для доступа к хранилищу (текстовый файл с паролем в первой строке)	Нет	
graphql.client.oauth.basic.tls.verify-host-name	Проверять ли серверный сертификат на соответствие имени хоста или IP-адреса	Нет	
oauth.basic.*	Глобальные свойства авторизации. Будут проигнорированы при наличии graphql.client.oauth.*	Нет	false

Пример конфигурации для заведения graphql клиента в листинге ниже.

Листинг 4 Пример конфигурации клиентов jdbc

```
# Глобальные свойства авторизации. Они будут использованы при отсутствии
`graphql.client.oauth.basic`
oauth:
  basic: &global-auth
    enabled: true
    url: http://172.24.26.67/oauth2/token?grant_type=client_credentials
    username: global-username
    password: passssssssssss

graphql:
  client:
    enabled: true
    url: https://k8s.supercode.ru:3600/phd/ft-phd-graphql-router/graphql
    connect-timeout-millis: 100
    read-timeout-millis: 1000
    cache:
      enabled: true
      maximumSize: 128
      expireAfterWrite: 2048
      useWeakReference: true
    oauth:
      # Эти свойства приоритетнее `graphql.client.oauth.basic`
      basic:
        <<: *global-auth
        enabled: true
        username: graph-username
        password: graph-pass
```

4.2.2.4 Конфигурация jdbc-модуля СИП

Модуль алгоритмов DQF-engine, предоставляющий общие механизмы взаимодействия JDBC. Драйвера не являются частью библиотеки и должны быть добавлены отдельно (см. прменение).

При выполнении проверок предусмотрена организация jdbc-клиентов, это является лучшей практикой использользования jdbc в DQF, т. к. соединения настраиваются пулами и не требуется передача кредов в тексте алгоритмов.

Настройки вида `engine.algo.jdbc.*` содержит список клиентов jdbc, для использования в алгоритме jdbc.

Таблица 16. Настройки клиентов jdbc.

Параметр	Что делает	Обязательно	Значение по умолчанию
`engine.algo.jdbc.clients.*`	Содержит список клиентов для подключения по jdbc	Нет	
`engine.algo.jdbc.clients[i].name`	Уникальное имя клиента	Условно.Да	
`engine.algo.jdbc.clients[i].hikari.*`	Префикс для подраздела со стандартными настройками пула соединений Hikari CP.	Да	

`engine.algo.jdbc.clients[i].hikari.jdbcUrl`	Строка подключения JDBC к базе данных. Определяет тип СУБД, хост, порт и имя базы.	Да	
`engine.algo.jdbc.clients[i].hikari.username`	Имя пользователя для аутентификации в БД.	Да	
`engine.algo.jdbc.clients[i].hikari.password`	Пароль пользователя для аутентификации в БД.	Да	

Пример конфигурации для заведения jdbc клиентов в листинге ниже.

Листинг 5 Пример конфигурации клиентов jdbc

```
# Конфигурация стенда demo
engine:
  algo:
    jdbc:
      clients:
        - name: demo-postgres
          hikari:
            jdbcUrl: jdbc:postgresql://127.0.0.1:5432/jdbc-examples
            username: admin
            password: admin
            maximumPoolSize: 4
            minimumIdle: 0
            connectionTimeout: 10000
            idleTimeout: 30000
            maxLifetime: 60000
        - name: demo-mysql
          hikari:
            jdbcUrl: jdbc:mysql://127.0.0.1:3307/jdbc-examples
            username: user
            password: user-pass
            maximumPoolSize: 2
            minimumIdle: 0
            connectionTimeout: 10000
            idleTimeout: 30000
            maxLifetime: 60000
        - name: demo-oracle
          hikari:
            jdbcUrl: jdbc:oracle:thin:@//127.0.0.1:1521/FREEPDB1
            username: test
            password: admin
            maximumPoolSize: 2
            minimumIdle: 0
            connectionTimeout: 10000
            idleTimeout: 30000
            maxLifetime: 60000
```

4.2.3 Конфигурационные параметры сервиса каталога правил (СКП)

Общие параметры «Каталога» – СКП (раздел 4.2.1) требуют настройки профиля, очередей, отправки событий, базы данных, при необходимости авторизации клиента, аудита, так же может получать параметры от сервера конфигураций. Далее приводятся параметры специфичные для СКП.

Сервис может интегрироваться с Arenadata Catalog для управления правилами из интерфейса ADC.

Таблица 17. Интеграция с ADC.

Параметр	Что делает	Обязательно	Значение по умолчанию
`dqf.integration.adc.enabled`	Флаг включения интеграции с ADC	Нет	`false`
`dqf.integration.adc.http-client.url`	Базовый URL ADC	Да, если интеграция включена	-
`dqf.integration.adc.http-client.connect-timeout-millis`	Время ожидания соединения	Нет	Задано в ReactorClientHttpConnector
`dqf.integration.adc.http-client.read-timeout-millis`	Время ожидания ответа на запрос	Нет	Задано в ReactorClientHttpConnector
`dqf.algorithm.max-request-size.add`	Максимально допустимый размер тела запроса в байтах для операции добавления алгоритма. Если запрос превышает этот размер, обработка прекращается с ошибкой 422.	Нет	`2MB`

Параметр	Что делает	Обязательно	Значение по умолчанию
dqf.algorithm.max-request-size.update	Максимально допустимый размер тела запроса в байтах для операции обновления алгоритма. Если запрос превышает этот размер, обработка прекращается с ошибкой 422.	Нет	2MB

4.2.4 Конфигурационные параметры сервиса планирования задач (СПЗ)

Общие параметры «Планировщика» СПЗ (раздел 4.2.1) требуют настройки профиля, очередей, отправки событий, базы данных, при необходимости авторизации технической учётной записи (ТУЗ), авторизации клиента, аудита, так же может получать параметры от сервера конфигураций. Далее приводятся параметры специфичные для СПЗ.

Таблица 18. Общие конфигурационные параметры специфичные для СПЗ.

Параметр	Описание	Обязательно	Значение по умолчанию
spring.cache.caffeine.spec	Задействовать механизм кэширования для групп правил по идентификатору и версии.	Нет	expireAfterWrite=PT2H,maximumSize=100
dqf.task.upload.max-request-size	Максимально допустимый размер тела запроса в байтах. Если запрос превышает этот размер, обработка прекращается с ошибкой 422.	Нет	2MB

Сервис интегрируется с Arenadata Catalog для синхронизации проверок и результатов. А именно может запускать проверки из интерфейса ADC.

Таблица 19. Интеграция с ADC.

Параметр	Что делает	Обязательно	Значение по умолчанию
dqf.integration.adc.enabled	Флаг включения интеграции с ADC	Нет	false
dqf.integration.adc.http-client.url	Базовый URL ADC	Да, если интеграция включена	-
dqf.integration.adc.http-client.connect-timeout-millis	Время ожидания соединения	Нет	Задано в ReactorClientHttpConnector
dqf.integration.adc.http-client.read-timeout-millis	Время ожидания ответа на запрос	Нет	Задано в ReactorClientHttpConnector

СПЗ интегрируется с каталогом правил (СКП) при помощи параметров, описанных ниже.

Таблица 20. Конфигурационные параметры интеграции с каталогом правил.

Параметр	Описание	Обязательно	Значение по умолчанию
dqf.planner.cas-service.url	URL Каталога Проверок для получения информации о Группе алгоритмов	Да	

Сервис осуществляет отправку сообщений в СИП. Сообщения отправляются пачками. Решение об отправке очередной пачки сообщений принимается на основе опроса очереди.

Таблица 21. Конфигурационные параметры интеграции с СИП

Параметр	Описание	Обязательно	Значение по умолчанию
dqf.planner.engine.enabled	Включение интеграции с DQF ES. **Значение false отключает все остальные конфигурации интеграции**.	Нет	true
dqf.planner.engine.queue	Очередь/топик чтения DQF ES. В нее будет осуществляться отправка сообщений.	Да	
dqf.planner.engine.buffer-size	Размер буфера сообщений, отправляемых в DQF Engine.	Нет	10000
dqf.planner.engine.lag-poll-rate-ms	Интервал опроса очереди(топика) DQF ES в миллисекундах. Определяет частоту опроса очереди (и попытке отправить следующую пачку сообщений).	Нет	100
dqf.planner.engine.message-send-delay	Задержка между появлением сообщения для отправки и отправкой сообщения в очередь. Duration, не отрицательное. Должно быть больше spring.transaction.default-timeout.	Нет	PT30S
dqf.planner.ongoing-task-cancellation.page-size	Размер страницы для получения просроченных Задач. Положительное число.	Нет	500
dqf.planner.ongoing-task-cancellation.expired-tasks-poll-rate-ms	Интервал опроса БД в миллисекундах. Определяет частоту опроса БД с целью определения "просроченных" Задач. Не может быть меньше, чем dqf.planner.ongoing-task-cancellation.expiration-timeout	Нет	1000
dqf.planner.ongoing-task-cancellation.expiration-timeout	Таймаут на загрузку данных в рамках Ongoing Задачи. Duration, не отрицательное.	Нет	PT5M
dqf.planner.ongoing-task-cancellation.user-cancel	Пользователь, от чьего имени происходит автоматическая отмена Ongoing-Задачи.	Нет	planner-backend
spring.transaction.default-timeout	Таймаут транзакции. Должен быть меньше задержки dqf.planner.engine.message-send-delay.	Нет	PT25S
spring.datasource.hikari.maximum-pool-size	Количество подключений к БД.	нет	10

Помимо общей конфигурации требуется настройка профиля:

- dqf-kafka - для работы с DQF ES по Kafka (см. Таблица 22)
- dqf-rmq - для работы с DQF ES по RMQ. **Профили являются взаимоисключающими**.

Таблица 22. Конфигурационные параметры интеграции с СИП по Kafka – профиль dqf-kafka

Параметр	Описание	Обязательно	Значение по умолчанию
dqf.planner.engine.kafka.group-id	Название Группы Потребителей DQF ES. Необходимо для получения лага обработки.	Да	
spring.kafka.*	Настройка подключения к Kafka. В рамках интеграции будет создан admin-клиент.	Да	
spring.kafka.producer.key-serializer	Сериализатор ключей.	Нет	org.apache.kafka.common.serialization.StringSerializer
spring.kafka.producer.value-serializer	Сериализатор значений.	Нет	org.apache.kafka.common.serialization.ByteArraySerializer

Конфигурационные параметры интеграции с СИП по RabbitMQ – профиль dqf-rmq

Конфигурация производится стандартной настройкой spring.rabbitmq.*. Свойство dqf.planner.engine.queue выступает как алиас к свойству spring.rabbitmq.template.routing-key

4.2.5 Конфигурационные параметры сервиса расписаний задач (СРЗ)

Общие параметры СРЗ (раздел 4.2.1) требуют настройки профиля, очередей, базы данных, отправку событий, при необходимости авторизации технической учётной записи (ТУЗ), авторизации клиента, аудита, так же может получать параметры от сервера конфигураций. Далее приводятся параметры специфичные для СРЗ.

Таблица 23. Конфигурационные параметры интеграций с сервисами DQF

Параметр	Описание	Обязательно	Значение по умолчанию
`dqf.scheduler.enabled`	Флаг включение создания Задач в Планировщике по Расписанию.	Нет	`true`
`dqf.scheduler.planned-polling-period`	Частота опроса Расписаний, готовых к отправке в Планировщик. Cron-выражение.	Нет	`0 * * * * *`
`dqf.scheduler.parallelism`	Доступное количество параллельных потоков для создания Задач по Расписанию. 1-255	Нет	`1`
`dqf.scheduler.planner-service.url`	URL Планировщика для создания Задач	Да	
`dqf.scheduler.cas-service.url`	URL Каталога Проверок для получения информации о Группе алгоритмов	Да	

4.2.6 Конфигурационные параметры сервиса результатов («Агрегатор»)

Общие параметры «Агрегатора» – сервиса результатов (раздел 4.2.1) требуют настройки профиля, очередей, отправки событий, базы данных, при необходимости авторизации клиента, аудита, так же может получать параметры от сервера конфигураций. Далее приводятся параметры специфичные для «Агрегатора».

Сервис интегрируется с Arenadata Catalog для синхронизации результатов проверок и передачи формирования событий завершения проверки с результатами.

Таблица 24. Интеграция с ADC.

Параметр	Что делает	Обязательно	Значение по умолчанию
`dqf.integration.adc.enabled`	Активировать ли обработку событий для ADC	Нет	false
`dqf.integration.adc.dqf-events-topic`	Топик, в котором слушатель событий для ADC будет ожидать новые события. По-умолчанию совпадает с очередью, для отправки событий.	Условно	${dqf.events.queue}
`dqf.integration.adc.http-client.url`	Базовй URL сервиса ADC, куда будет отправляться обработанное событие	Условно	-
`dqf.integration.adc.http-client.connect-timeout-millis`	Время ожидания соединения	Нет	Задано в ReactorClientHttpConnector
`dqf.integration.adc.http-client.read-timeout-millis`	Время ожидания ответа на запрос	Нет	Задано в ReactorClientHttpConnector

Таблица 25. Интеграция с СИП.

Параметр	Что делает	Обязательно	Значение по умолчанию
`dqf.aggregator.engine.enabled`	Флаг для выключения обработки результатов работы движка. Остальные свойства не актуальны, если обработка выключена		true

Параметр	Что делает	Обязательно	Значение по умолчанию
dqf.aggregator.engine.result-queues	Названия очередей/топиков, разделенных запятой, в которую DQF ES отправляет результаты выполнения проверок. Например: success_q,fail_q,error_q	Да	
dqf.aggregator.events.enabled	Флаг для выключения обработки событий. Остальные свойства событий не актуальны, если обработка выключена	Да	true
dqf.aggregator.events-queue	Очередь/топик из которой сервис должен читать события	Да	

4.2.7 Конфигурационные параметры сервиса дашбордов

Общие параметры сервиса дашбордов (раздел 4.2.1) требуют настройки профиля, базы данных, при необходимости авторизации технической учётной записи (ТУЗ), авторизации клиента, аудита, так же может получать параметры от сервера конфигураций. Параметры специфичные для сервиса дашбордов отсутствуют.

4.2.8 Конфигурационные параметры сервиса отчетов

Общие параметры сервиса отчётов (раздел 4.2.1) требуют настройки профиля, базы данных, при необходимости авторизации технической учётной записи (ТУЗ), авторизации клиента, аудита, так же может получать параметры от сервера конфигураций. Параметры специфичные для сервиса отчётов далее (Таблица 26)

Таблица 26. Параметры сервиса отчётов

Параметр	Что делает	Обязательно	Значение по умолчанию
dqf.reporter.concurrency	Количество параллельных потоков для создания отчетов	Нет	2
dqf.reporter.process-health-period	Периодичность обновления метки времени для отчетов в обработке. Также отчеты, метка которых не обновлялась 3 таких периода, возвращаются в очередь	Нет	60000
dqf.reporter.report-polling-period	Период опроса очереди шаблонов	Нет	200

Так же сервис интегрируется с Сервисом результатов – «Агрегатором». Настройки интеграции приведены ниже

Таблица 27. Параметры сервиса отчётов

Параметр	Что делает	Обязательно	Значение по умолчанию
dqf.reporter.aggregator.url	URL Аггрегатора для получения данных для отчетов		Да

4.2.9 Конфигурационные параметры сервиса авторизации

Общие параметры сервиса авторизации (раздел 4.2.1) требуют настройки профиля, базы данных, при необходимости авторизации клиента, аудита, так же может получать параметры от сервера конфигураций.

Настройка авторизации через внешний Keycloak-сервис. Все защищенные ресурсы будут ожидать, что пользователь совершает запросы с авторизационным заголовком `Authorization: Bearer <JWT from Keycloak>`

Таблица 28. Параметры получения атрибутов доступа

Параметр	Описание
`spring.security.oauth2.web.*`	Необходимые для OIDC-аутентификации параметры для web UI
`spring.security.oauth2.web.provider`	Провайдер OIDC: `keycloak` или `avanpost` (По умолчанию: `keycloak`)
`spring.security.oauth2.web.url`	Ссылка на сервер OIDC для web UI
`spring.security.oauth2.web.realm`	Реалм для аутентификации (обязателен для провайдера `keycloak`)

Таблица 29. Параметры получения атрибутов доступа

Параметр	Что делает	Обязательно	Значение по умолчанию
`dqf.auth.attribute-provider.type`	Тип провайдера атрибутов доступа. Допустимые значения: [TOKEN, ADC]	Нет	TOKEN *Примечание:* TOKEN означает, что атрибуты доступа будут получены из токена. ADC – сервис получает атрибуты доступа REST-запросом на внешний провайдер (см. ниже Таблица 31)
`spring.cache.caffeine.spec`	Спецификация Caffeine-кэша для получения атрибутов доступа.	Нет	expireAfterWrite=PT10S,maximumSize=100,recordStats

Получение атрибутов доступа из JWT-токена: dqf.auth.attribute-provider.type=TOKEN

Таблица 30. Параметры получения атрибутов из JWT-токена

Параметр	Что делает	Обязательно	Значение по умолчанию
`spring.security.oauth2.client.provider.keycloak.dqf-authority-attribute`	Свойство JWT-токена, в котором хранятся атрибуты DQF.	Нет	DQF_ACCESS_ATTRIBUTE

Получение атрибутов доступа из JWT-токена: dqf.auth.attribute-provider.type=ADC

Таблица 31. Параметры получения атрибутов из JWT-токена

Параметр	Что делает	Обязательно	Значение по умолчанию
`dqf.auth.attribute-provider.adc.url`	Ссылка на внешний провайдер атрибутов - ADC.	-	Да

4.2.10 Конфигурационные параметры адаптера результатов для модуля сохранения

Общие параметры сервиса авторизации (раздел 4.2.1) требуют настройки профиля, очередей, базы данных, при необходимости авторизации технической учётной записи (ТУЗ), аудита, так же может получать параметры от сервера конфигураций. Далее приводятся параметры специфичные для адаптера результатов.

Таблица 32. Обязательные параметры

Параметр	Что делает	Обязательно	Пример
`dqf.saver.adapter.engine-topics`	Названия очередей/топиков, разделенных запятой, в которую DQF ES отправляет результаты выполнения проверок.	Да	success_q,fail_q,error_q
`dqf.saver.adapter.event-topic`	Названия топика, в которую отправляются события системы DQF.	Да	success_q,fail_q,error_q
`dqf.saver.adapter.saver-topics.success`	Название очереди для отправки успешных результатов в Results Saver.	Да	result_success
`dqf.saver.adapter.saver-topics.failed`	Название очереди для отправки неуспешных результатов в Results Saver.	Да	result_failed
`dqf.saver.adapter.saver-topics.exception`	Название очереди для отправки результатов, завершившихся ошибкой, в Results Saver.	Да	result_error
`dqf.saver.adapter.saver-topics.signals`	Название очереди для отправки сигналов в Results Saver.	Да	signals-topic
`dqf.saver.adapter.task-id-header-name`	Название заголовка для передачи Task Id в Results Saver.	Да	pipilineRunId

4.3 Настройка пользователей

4.3.1 OIDC-провайдер (Keycloak/Avanpost)

Ролевая модель определяется атрибутами доступа, определяемыми атрибутами авторизации в JWT токене выданном OIDC-провайдером.

Атрибуты авторизации по умолчанию DQF_ACCESS_ATTRIBUTE определяют права пользователя.

Группы прав определяют роль, т. е. каждой группе назначаются множество значений по ключу DQF_ACCESS_ATTRIBUTE. Допустимые значения DQF_ACCESS_ATTRIBUTE:

1. "RULE_CREATE" – создать правило,
2. "RULE_READ" – прочитать правило,
3. "RULE_UPDATE" – обновить правило,
4. "RULE_DELETE" – удалить правило,
5. "RULE_PROBE" – запустить правило в режиме дебага,
6. "RULE_GROUP_CREATE" – создать группу правил,
7. "RULE_GROUP_READ" – прочитать группу правил,
8. "RULE_GROUP_UPDATE" – обновить группу правил,
9. "RULE_GROUP_DELETE" – удалить группу правил,
10. "TASK_CREATE" – создать задачу,
11. "TASK_READ" – прочитать задачу,
12. "TASK_UPDATE" – обновить задачу,
13. "SCHEDULE_CREATE" – создать расписание,
14. "SCHEDULE_READ" – прочитать расписание,
15. "SCHEDULE_UPDATE" – обновить расписание,
16. "REPORT_CREATE" – создать отчёт,
17. "REPORT_READ" – прочитать отчёт,
18. "REPORT_UPDATE" – обновить отчёт,
19. "RESULT_READ" – прочитать результат,

20. "AGGREGATE_STATISTIC_READ" – прочитать агрегированную статистику результатов,
21. "DASHBOARD_CREATE" – создать дашборд,
22. "DASHBOARD_READ" – прочитать дашборд,
23. "DASHBOARD_UPDATE" – обновить дашборд,
24. "DASHBOARD_DELETE" – удалить дашборд

Атрибуты назначаются группам пользователей, в стандартной версии DQF выделяется 9 ролей:

- Пользовательские (5 ролей – без выделения)
  - Владелец данных (data-owner)
  - Менеджер по качеству данных (manager)
  - Аналитик качества данных (data-analyst)
  - Разработчик правил (rules-developer)
  - Дата-стюард (data-steward)
- Технические (4 роли – выделены жёлтым)
  - Сервис отчётов (reporter)
  - Сервис расписаний (schedule)
  - Сервис исполнения проверок (dqf)
  - Скрипт обновления правил для каталога (cas-updater)

Таблица 33. Маппинг атрибутов на роли (группы пользователей Keycloak)

Роль	Атрибуты																								
	RULE					RULE_GROUP				TASK			SCHEDULE			REPORT			Results		DASHBOARD				
	C	R	U	D	P	C	R	U	D	C	R	U	C	R	U	C	R	U	1	∑	C	R	U	D	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
data-owner		v					v				v			v			v		v	v		v			
manager		v					v				v			v			v		v	v		v			
data-analyst		v					v				v			v			v		v	v	v	v	v	v	
rules-developer	v	v	v	v	v					v	v	v							v						
data-steward		v				v	v	v	v	v	v	v	v	v	v	v	v	v	v	v		v			
reporter																			v	v					
schedule										v	v	v													
dqf		v																							
cas-updater		v				v		v	v																

Для удобства тёмно-зелёным выделены полные группы, светло-зелёным – часть группы.

4.3.2 Конфигурация пользователей при интеграции с ADC

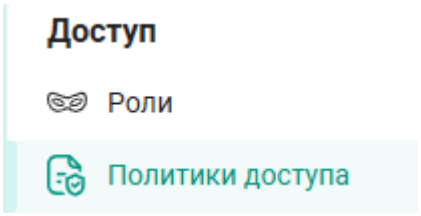
Ролевая модель определяется политиками доступа.

Добавление политик связанных с DQF осуществляется в настройках Arenadata catalog –  , раздел меню «Доступ», пункт «Политики доступа» (см. Рисунок 2, а).

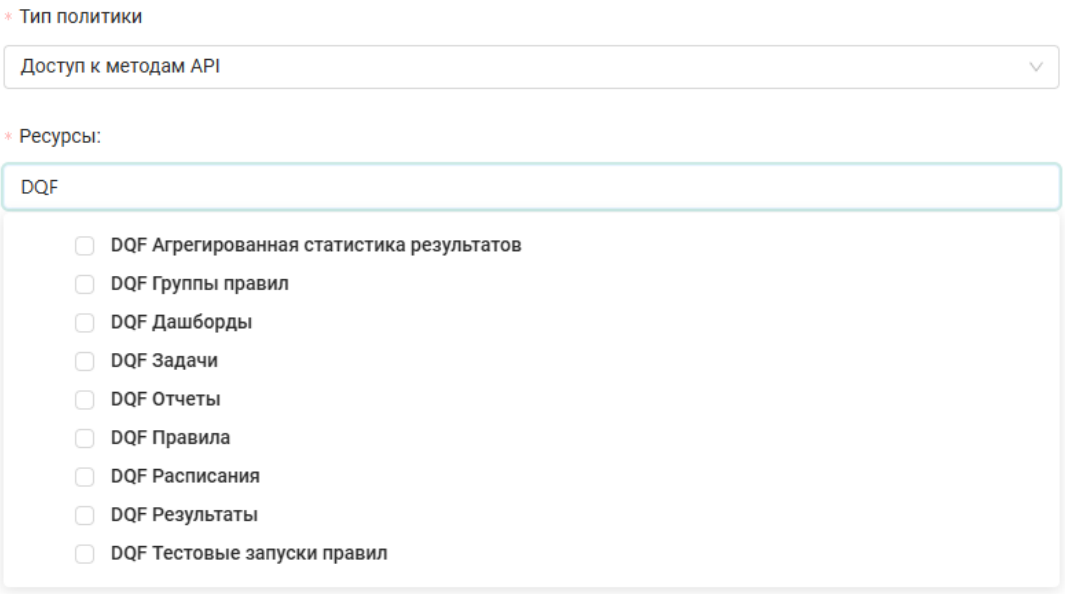
При отсутствии существующих политик связанных с DQF – Создать новую политику с типом «Доступ к методам API», ресурсом, одним из списка (см. Рисунок 2, б) (интерфейс позволяет искать по сочетанию символов «DQF»), после чего надо выбрать допустимую операцию, применимую к ресурсу (см. Рисунок 2, б). Данные типы операций соответствуют 24 группам прав (см. раздел 4.3.1).

После добавления политики, данная политика может применяться в правах доступа к командам, пользователям или ролям, по стандартному процессу ADC)

а)



б)



в)

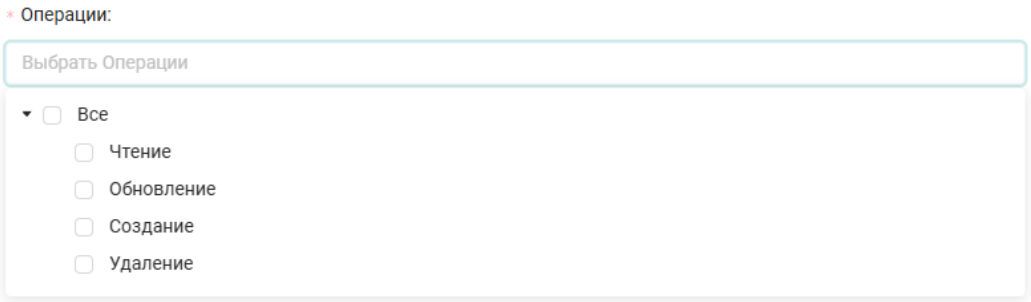


Рисунок 2. Архитектурная схема ADC.DQF

4.4 Настройка сбора метрик

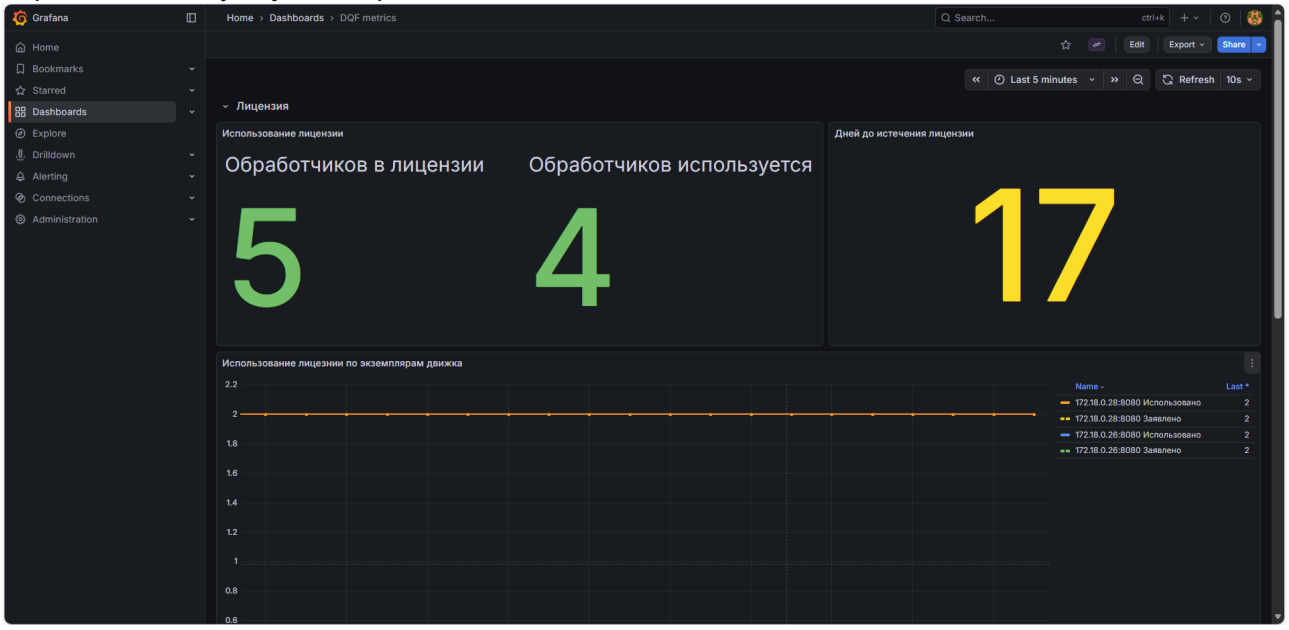
Метрики контейнеров в формате Prometheus доступны по адресу:
{хост-сервиса:порт-сервиса}/actuator/prometheus

Перечень сервисов, их размещение и информация с административными доступами формируется при инициализации стендов и размещается в каталоге стенда в файле {stand_name}-stand.md

4.5 Просмотр состояния системы

При типовом развёртывании ADC.DQF на стенде http://{хост}:3000/ располагается Grafana, в которой можно увидеть состояние лицензии, при необходимости обновить лицензию можно

обратиться в службу поддержки.



Так же там могут настраиваться для наблюдения другие метрики системы

5 Аварийные ситуации

В настоящем разделе приведён перечень возможных неисправностей, причины их возникновения и способы устранения.

В случае появления иных неисправностей следует обратиться к поставщику ПО (см. раздел 7, стр. 38)

Таблица 34 Перечень возможных неисправностей

№ п/п	Неисправность	Возможная причина	Способ устранения
1	Недоступен пользовательский интерфейс	Сервис веб-доступа (nginx) не запущен	Проверить состояние сервиса nginx и выполнить его запуск (см. раздел 3.3 стр. 10 имя контейнера по умолчанию **nginx-reverse-proxy**)
2	Невозможна авторизация пользователя	Некорректная настройка **realm** в системе авторизации	Проверить состояние сервиса Keycloak и корректность настройки **realm** (см. раздел 3.3 стр. 10 имя контейнера по умолчанию **keycloak**, конфигурации)
3	Недоступен тестовый запуск правила	Не установлен экземпляр СИП с параметром PROBE (см. раздел 4.2.2)	Проверить установку и состояние экземпляра СИП – **dqf-es-probe** (см. раздел 3.3 стр. 10 имя контейнера по умолчанию **dqf-es-probe**)
		Отсутствует или недействительна лицензия	Проверить наличие и срок действия лицензии (см. раздел 4.5 стр. 34)
4	Задача запущена, но не выполняется	Сервис исполнения правил (СИП) не функционирует	Проверить состояние СИП – **dqf-es** (см. раздел 3.3 стр. 10 имя контейнера по умолчанию **dqf-es**)
		Ограничение лицензии	Проверить лицензию на актуальность и доступность потоков экземпляру СИП (см. раздел 4.5 стр. 34)
5	Правила не отображаются в каталоге правил	Скрипт загрузки демонстрационных правил не выполнен	Запустить выполнение скрипта загрузки демонстрационных правил. Для чего необходимо стандартными средствами docker запустить или перезапустить контейнер **cas-updater-keycloak**.

6 Рекомендации по освоению

Для успешной работы с СИП необходимо изучить эксплуатационную документацию, которая представлена в разделе 1.5.

7 Контакты технических специалистов

В случае возникновения трудностей при установке программного обеспечения, свяжитесь с технической поддержкой, используя электронный адрес info@arenadc.io.