ИНФРАСТРУКТУРА ЭЛЕКТРОННОГО ПРАВИТЕЛЬСТВА

Руководство администратора

Типового программного обеспечения управления данными

Версия 1.0.1

**СМЭВ4**

Листов 36

Москва, 2025

Содержание

[1 Введение 5](#_Toc201145945)

[1.1 Область применения 5](#_Toc201145946)

[1.2 Краткое описание возможностей 5](#_Toc201145947)

[1.3 Уровень подготовки пользователя 5](#_Toc201145948)

[2 Назначение и условия применения 6](#_Toc201145949)

[2.1 Назначение 6](#_Toc201145950)

[2.2 Требования к среде для развертывания ТПО УД 6](#_Toc201145951)

[2.2.1 Минимальные требования к серверу 6](#_Toc201145952)

[2.2.2. Требования к установке и подключению Агента СМЭВ4 6](#_Toc201145953)

[2.2.3 Перечень стороннего ПО, необходимого для работы ТПО УД 6](#_Toc201145954)

[2.3 Точки подключения к ТПО УД 6](#_Toc201145955)

[3 Настройка и запуск ТПО УД 7](#_Toc201145956)

[3.1 Порядок загрузки данных и программ 7](#_Toc201145957)

[3.2 Подготовка и настройка системы для запуска ТПО УД 7](#_Toc201145958)

[3.2.1 Состав и содержание дистрибутивного пакета 7](#_Toc201145959)

[3.2.2 Предварительные операции (установка «пре-реквизитов») 7](#_Toc201145960)

[3.2.2.1 Общие предварительные действия 7](#_Toc201145961)

[3.2.2.2 Настройка схемы датамарта качества (dqf) 8](#_Toc201145962)

[3.2.3 Запуск ТПО УД 8](#_Toc201145963)

[3.2.4 Настройка ротации логов (опционально) 9](#_Toc201145964)

[4 Описание конфигурационных файлов ТПО УД 11](#_Toc201145965)

[4.1 Состав и содержание конфигурационных файлов 11](#_Toc201145966)

[4.2 Реестр изменений файлов конфигурационных 11](#_Toc201145967)

[4.3 Формирование конфигурационного файла ТПО УД 11](#_Toc201145968)

[4.3.1 Настройка подключения к сервису аутентификации 11](#_Toc201145969)

[4.3.2 Настройка параметров подключения к Ядру СМЭВ4 11](#_Toc201145970)

[4.3.3 Настройка использования альтернативного значения серверного порта 11](#_Toc201145971)

[4.3.4 Настройка параметров Витрин 12](#_Toc201145972)

[4.3.5 Настройка параметров передачи данных аудита гостех 12](#_Toc201145973)

[4.3.6 Настройка ТПО УД для передачи метрик в Prometheus 12](#_Toc201145974)

[4.3.7 Настройка количества одновременных соединений 12](#_Toc201145975)

[4.3.8 Настройка максимального количества строк в CSV файле 12](#_Toc201145976)

[4.4 Настройка конфигурационного файла nginx.conf 13](#_Toc201145977)

[Приложение А. Описание метрик 14](#_Toc201145978)

[Приложение Б. Описание спецификации для взаимодействия интерфейсов ТПО УД 18](#_Toc201145979)

[Датамарты 18](#_Toc201145980)

[Получение данных для выбора датамарта 18](#_Toc201145981)

[Получение данных таблиц датамарта 18](#_Toc201145982)

[Получение данных справочника проверок 19](#_Toc201145983)

[Получение результатов проверок (Ошибки в данных) 19](#_Toc201145984)

[Получение детальной информации о записи основной таблицы 21](#_Toc201145985)

[Выгрузка результатов проверок (Ошибки в данных) в формате CSV 22](#_Toc201145986)

[Получение детальной информации о проверке 23](#_Toc201145987)

[Получение статистики по выполненным проверкам 25](#_Toc201145988)

[Получение статистики по инцидентам 26](#_Toc201145989)

[Получение статистики по строкам и ошибкам 26](#_Toc201145990)

[Получение статистики ошибок по таблицам 27](#_Toc201145991)

[Произвольный SQL-запрос 28](#_Toc201145992)

[Инциденты 30](#_Toc201145993)

[Получение данных об инциденте 30](#_Toc201145994)

[Получение списка инцидентов 30](#_Toc201145995)

[Пользователи 32](#_Toc201145996)

[Получение данных о пользователе 32](#_Toc201145997)

[Выход пользователя из ТПО УД 32](#_Toc201145998)

[Ошибки 33](#_Toc201145999)

[Термины и определения 34](#_Toc201146000)

[История изменений документа 36](#_Toc201146001)

АННОТАЦИЯ

В данном программном документе приведено Руководство администратора Типового программного обеспечения управления данными (ТПО УД) – механизма контроля качества данных, размещённых на программном обеспечении «Витрина данных», через Единую систему межведомственного электронного взаимодействия версии 3.х.

В разделе 1 «Введение» указаны область применения подсистемы, краткое описание ее возможностей, уровень необходимой подготовки пользователей подсистемы и перечень документации для работы с ТПО УД.

В разделе 2 «Назначение и условия применения» описано назначение подсистемы и требования к среде развертывания.

В разделе 3 «Настройка и запуск ТПО УД» приведено описание порядка подготовки, настройки и запуска ТПО УД.

В разделе 4 «Настройка и запуск ТПО УД» приведено описание порядка подготовки, настройки и запуска ТПО УД.

В приложении А приведены описания метрик, используемых при работе с ТПО УД.

В приложении Б приведена спецификация для взаимодействия интерфейсов ТПО УД.

В разделе «Термины и определения» приведены используемые в данном Руководстве термины и их определения.

В разделе «История изменений» приведена таблица с изменениями данного документа.

# 1 Введение

## 1.1 Область применения

Типовое программное обеспечение управления данными (далее – ТПО УД) используется для проверки качества данных, размещенных в Типовом программном обеспечении «Витрина данных».

## 1.2 Краткое описание возможностей

ТПО УД обеспечивает:

* возможность просмотра реестра проверок качества данных;
* возможность просмотра журнала с результатами проверок качества данных;
* возможность просмотра агрегированной статистики качества данных;
* возможность просмотра атрибутов записи основной таблицы, не прошедших проверку.

## 1.3 Уровень подготовки пользователя

Развертывание ТПО УД должно осуществляться в соответствии с настоящим Руководством администратора.

Специалист должен обладать следующими знаниями и опытом:

* администрирование ОС Linux, версии применяемой для развертывания ТПО УД.

# 2 Назначение и условия применения

## 2.1 Назначение

ТПО УД – механизм контроля качества данных, размещённых на программном обеспечении «Витрина данных», через Единую систему межведомственного электронного взаимодействия версии 3.х.

## 2.2 Требования к среде для развертывания ТПО УД

### 2.2.1 Минимальные требования к серверу

[Таблица 1](#_8a6d1d8e05256c19e19f11d91ca90680) содержит минимальные требования к характеристикам сервера для развертывания ТПО УД.

Таблица 1 - Минимальные требования к характеристикам сервера для развертывания ТПО УД

|  |  |
| --- | --- |
| Характеристики | ТПО УД |
| ОС | * + ALT 8 SP Server 10;
	+ RedOS 7.3;
	+ Astra Linux 1.7 SE.
 |
| ЦПУ | 2-х ядерный процессор |
| Оперативная память | 4 Гб |
| Дисковое пространство | 20 Гб |

### 2.2.2. Требования к установке и подключению Агента СМЭВ4

Для работы ТПО УД необходимо выполнить установку Агента СМЭВ4 и подключение к нему. См. РА Агента СМЭВ4. Между ТПО УД и Агентом СМЭВ4 должна быть обеспечена сетевая связанность.

### 2.2.3 Перечень стороннего ПО, необходимого для работы ТПО УД

* + JDK версии 17.0, рекомендуется JDK 17 LTS Axiom Сертифицированный дистрибутив JDK, cоответствующий операционной системе, приобретается пользователем самостоятельно на сайте производителя <https://axiomjdk.ru/pages/downloads/#/java-17-lts>.
	+ Nginx версии 1.12.2 или выше;
	+ для отображения вкладки «Произвольный SQL-запрос» необходим доступ в СМЭВ4 к произвольным запросам, в случае если у ТПО УД Агент СМЭВ4 отличный от Агента Витрины;
	+ для отображения вкладки «Реестр проверок» должен быть настроен доступ к спецификациям ЕИП и Prostore Витрины.

## 2.3 Точки подключения к ТПО УД

[Таблица 2](#_87c82fc79b8df105548e3fef0d53c1b6) содержит точки подключения к ТПО УД и их назначение

Таблица 2 - Точки подключения к ТПО УД

|  |  |
| --- | --- |
| Порт (значение по умолчанию) [[1]](#footnote-1) | Назначение |
| 80 | Веб-интерфейс ТПО УД |

# 3 Настройка и запуск ТПО УД

## 3.1 Порядок загрузки данных и программ

Администратор УВ осуществляет развёртывание, запуск и настройку ТПО УД с помощью данного руководства.

## 3.2 Подготовка и настройка системы для запуска ТПО УД

### 3.2.1 Состав и содержание дистрибутивного пакета

Состав дистрибутива ТПО УД в архиве с бинарными файлами (/distr/ds):

* + /distr/ds/app/app.jar – исполняемый файл;
	+ /distr/ds/application.yml – файл конфигурации;
	+ /distr/ds/webapps/\* - файлы фронтенда;
	+ /distr/ds/nginx.conf – файл конфигурации веб-сервера Nginx.

Путь /distr/ds указан в качестве примера. Допустимо использовать любой другой путь, скорректировав соответствующим образом упоминаемые ниже команды.

Необходимо приобрести и загрузить с сайта <https://axiomjdk.ru/pages/downloads/#/java-17-lts> Axiom JDK 17-й версии, соответствующую используемой ОС.

### 3.2.2 Предварительные операции (установка «пре-реквизитов»)

#### 3.2.2.1 Общие предварительные действия

Перед запуском ТПО УД выполнить от имени пользователя **root** либо используя команду sudo следующие действия:

1. Создать пользователя {{user}} и группу {{user\_group}}, под которым будет работать ТПО УД. Распаковать дистрибутивный пакет в каталог /distr/ds
2. Установить JDK (версия в примере ниже может отличаться для загруженного пакета)

Для ALT 8 SP Server 10

apt install axiomjdk-jdk-certified17.0.6+11-linux-amd64.alt8.rpm

Для RedOS 7.3

yum install axiomjdk-jdk-certified17.0.6+11-linux-amd64.rpm

Для Astra Linux 1.7 SE

apt-get install axiomjdk-jdk-certified17.0.6+11-linux-amd64.deb

1. Установить системные переменные

export JAVA\_HOME=/usr/lib/jdk/axiomjdk-java17.x86\_64
export PATH=$PATH:$JAVA\_HOME/bin

Переменные должны устанавливаться автоматически также после перезагрузки сервера.

Для этого рекомендуется внести данную настройку в **profile** ОС (/etc/profile.d).

1. Выложить в каталог /distr/ds конфигурационный файл application.yml, подготовленный согласно описанию в [Разделе 4](#_4_Описание_конфигурационных).
2. Установить веб-сервер Nginx из репозитория используемой ОС

Для ALT 8 SP Server 10

apt install nginx

Для RedOS 7.3

yum install nginx

Для Astra Linux 1.7 SE

apt-get install nginx

1. Выложить в каталог /etc/nginx конфигурационный файл nginx.conf, подготовленный согласно описанию в [Разделе 4](#_4_Описание_конфигурационного).
2. Создать каталог /usr/webapps, скопировать в него все содержимое каталога /distr/ds/webapps дистрибутивного пакета, с сохранением файловой структуры.

#### Настройка схемы датамарта качества (dqf)

Перед настройкой и запуском системы "ТПО УД" необходимо выполнить следующие шаги, связанные с датамартом качества (dqf):

1. Настройка системы "Агент Проверок":

В конфигурационном файле системы "Агент Проверок" (application.yml Агента Проверок) необходимо убедиться, что задано указание схемы для датамарта качества. Пример конфигурации:

dqf:

 # Явное указание схемы для dqf

 schema: ${DQF\_SCHEMA:**reg\_datamart\_dqf**}

1. Запуск "Агента Проверок":

После выполнения настройки, необходимо убедиться, что "Агент Проверок" успешно запущен. При своем запуске он создаст датамарт с указанным наименованием **reg\_datamart\_dqf** и необходимые таблицы внутри него для хранения данных качества.

1. Настройка "ТПО УД":

Только после успешного запуска "Агента Проверок" и создания им датамарта качества, в конфигурационном файле "ТПО УД" (application.yml) необходимо задать датамарт качества с тем же наименованием, что было указано в конфиге "Агента Проверок":

datamart:

 list:

 - mnemonic: ac\_dc # мнемоника основного датамарта

 dqf: **reg\_datamart\_dqf** # мнемоника датамарта качества

Убедитесь, что значение dqf совпадает со значением schema в конфигурации "Агента Проверок".

### 3.2.3 Запуск ТПО УД

Убедитесь, что пользователь {{user}} имеет доступ к директории, в которой установлен ТПО УД (/distr/ds).

Пример раздела конфигурации скрипта для запуска ТПО УД с использованием сервиса **systemd**

Данный файл, с названием datasteward.service, необходимо сохранить в каталог /etc/systemd/system/, указав, при необходимости, корректные пути к файлам:

[Unit]
Description="Data steward AWP backend"
After=syslog.target

[Service]
Type=simple

WorkingDirectory=/distr/ds
LimitNOFILE=1048576:1048576

User={{user}} *# указать пользователя*
Group={{user\_group}} *# указать группу*

Environment="JAVA\_HOME=/usr/lib/jdk/axiomjdk-java17.x86\_64"
ExecStart=/usr/lib/jdk/axiomjdk-java17.x86\_64/bin/java -jar app/app.jar

[Install]
WantedBy=multi-user.target

Далее от имени пользователя **root** или используя sudo выполнить команду по включению, запуску сервиса и проверке его статуса:

sudo systemctl daemon-reload
sudo systemctl enable datasteward
sudo systemctl start datasteward
sudo systemctl -l status datasteward

Запустите сервис Nginx, убедитесь что он находится в состоянии Active (Running)

sudo systemctl enable nginx
sudo systemctl start nginx
sudo systemctl -l status nginx

### 3.2.4 Настройка ротации логов (опционально)

Во избежание переполнения диска лог-файлами, рекомендуется настроить ротацию лог-файлов **systemd** сервиса.

Настройка логирования осуществляется с помощью файлов конфигурации **logrotate** и **systemd** сервиса. Если этих файлов не существует, их необходимо создать.

Первоначально необходимо установить пакет **logrotate**, для rpm-base дистрибутивов необходимо выполнить команду:

sudo yum install logrotate

Добавьте в файл **systemd** сервиса (/etc/systemd/system/datasteward.service) следующие настройки логирования:

...
[Service]
Type=simple

StandardOutput=append:/var/log/{{ file\_name }}
StandardError=append:/var/log/{{ file\_name }}

...

Создайте файл конфигурации **logrorate** (поместить файл в /etc/logrotate.cron/datasteward.conf) и поместите в него следующие строки:

/var/log/{{ file\_name }} {
 rotate {{ rotate }}
 size {{ file\_size }}
 create
 nocompress
 copytruncate
}

где:

* rotate – какое количество лог файлов оставлять,
* file\_size – размер лог-файла при котором будет происходить ротация.

Создайте файл со скриптом запуска утилиты **logrotate**, поместите скрипт по пути /etc/logrotate.cron/datasteward\_logrotate.sh

*#!/bin/bash*
/usr/sbin/logrotate /etc/logrotate.cron/datasteward.conf

Добавьте права на запуск командой:

chmod +x /etc/logrotate.cron/datasteward.sh

Создайте правила cron, создайте файл /etc/cron.d/datasteward и поместите в него приведенную ниже строку:

\*/1 \* \* \* \* root /etc/logrotate.cron/datasteward\_logrotate.sh

Далее перезагрузите **systemd** сервис:

sudo systemctl daemon-reload
sudo systemctl restart datasteward

# 4 Описание конфигурационных файлов ТПО УД

## 4.1 Состав и содержание конфигурационных файлов

В дистрибутивном пакете находятся два конфигурационных файла:

* application.yml – конфигурация backend приложения;
* nginx.conf – конфигурация веб-сервера Nginx.

## 4.2 Реестр изменений файлов конфигурационных

[Таблица 3](#_1590441ee0ce88eec9ac24c6874db5a5) содержит Реестр изменений.

Таблица 3 - Реестр изменений конфигурационных файлов

| Версия | Перечень изменений (относительно предыдущей версии) |
| --- | --- |
| 1.0 | Конфигурационный файл создан |
| 1.0.1 | Изменений нет |

## 4.3 Формирование конфигурационного файла ТПО УД

### 4.3.1 Настройка подключения к сервису аутентификации

iam:

 enable: *true*

 host: *auth\_server\_address*

 realm: CustomersAuth

 path:

 user: map[user:/auth/realms/CustomersAuth/protocol/openid-connect/userinfo]

Где

* enable – включение/выключение аутентификации;
* host – адрес сервера аутентификации.

### 4.3.2 Настройка параметров подключения к Ядру СМЭВ4

agent:

 host: [http://*host:port*](http://host:port)

 datamart: #set yours datamart name

Где

* host – url для подключения к Агенту СМЭВ4;
* datamart – наименование ИС.

### 4.3.3 Настройка использования альтернативного значения серверного порта

Существует два варианта изменения значения по умолчанию номера порта:

1) Значение может быть передано через переменную окружения SERVER\_PORT. В случае запуска приложения с помощью systemd переменную необходимо передать через указание Environment="SERVER\_PORT=8080" в .service файле.

В случае прямого запуска, используя java -jar app/app.jar переменная должна быть задана на уровне ОС на момент запуска приложения/

2) Возможно указать в конфигурационном файле следующую настройку:

http-server:

 port: 8080

Где port – порт сервера, значение может быть изменено на нужное.

При изменении значения порта, необходимо изменить upstream в Nginx фронтенда соответственно.

### 4.3.4 Настройка параметров Витрин

datamart:

 mnemonic:

 list:

 dqf:

 default: demo\_acdc

Где

* + mnemonic – мнемоника datamart (основная схема);
	+ list – массив из пар;
	+ dqf – схема dqf-данных (датамарт качества);
	+ default – мнемоника по умолчанию.

### 4.3.5 Настройка параметров передачи данных аудита гостех

gostech:

 enabled: true

 audit:

 host: *audit-server-host*

 port: *audit-server-port*

 path-prefix: /v1

Где

* enable – включение/выключение передачи данных аудита
* host – адрес сервера аудита
* port – порт сервера аудита

### 4.3.6 Настройка ТПО УД для передачи метрик в Prometheus

metrics:

 implementation: PROMETHEUS

 endpointPort: 8381

Описание метрик приведено в [приложении А](#_Приложение_А._Описание).

### 4.3.7 Настройка количества одновременных соединений

vertx:

 web-client:

 max-pool-size: 20

### 4.3.8 Настройка максимального количества строк в CSV файле

Quality:

 csv-max-count: ${CSV\_MAX\_COUNT:1000}

## 4.4 Настройка конфигурационного файла nginx.conf

Данный файл рекомендуется использовать из дистрибутивного пакета, при необходимости в нем возможно скорректировать значения, выделенные ниже курсивом.

 server {

 listen 80;

 server\_name data-steward-portal;

 client\_max\_body\_size 20M;

 proxy\_set\_header Host $host;

 **# путь к файлу логов nginx**

 access\_log */var/log/nginx/access.log* main;

 gzip on;

 gzip\_static always;

 gzip\_comp\_level 5;

 gzip\_types

 application/javascript

 image/gif

 image/jpeg

 image/png

 image/tiff

 image/x-icon

 text/css;

 index index.html;

 **# здесь и далее, /usr/webapp – путь, по которому выложены файлы**

 **# фронтенда из дистрибутивного пакета**

 location / {

 root */usr/webapps*/ds-portal/;

 }

 location /assets {

 alias */usr/webapps*/assets/;

 try\_files $uri $uri/ /assets;

 }

 location /lib-assets {

 alias */usr/webapps*/lib-assets/;

 try\_files $uri $uri/ /lib-assets;

 }

 location /lkuv-lib-assets {

 alias */usr/webapps*/lkuv-lib-assets/;

 try\_files $uri $uri/ /lkuv-lib-assets;

 }

 location /api {

 **# адрес подключения в backend ТПО УД**

 **# в случае запуска фронтенд и бэкенд в пределах одного**

 **# сервера, использовать localhost**

 proxy\_pass http://*localhost:8080*/api;

 proxy\_redirect default;

 }

 }

# Приложение А. Описание метрик

Приложение ТПО УД формирует метрики в формате *Prometheus.*

Требования к сбору метрик указаны по ссылке:

<https://platform-docs.v-serv.ru/online-documentation/home/maintain-tools/monitoring/2.1.0/quick-start/>

Пример дашборда с jvm-метриками:

<http://10.81.7.90:29496/d/2YFG8xhVk1/jvm-metrics-pods?orgId=1&refresh=30s&var-kubernetes_namespace=podd-acc2&var-instance=einfahrt-consumer-1&var-jvm_memory_pool_heap=All&var-jvm_memory_pool_nonheap=All>

Пример дашборда с прикладными метриками:

<http://10.81.7.90:29496/d/i16VBBX4k1/agent-podd-gostech?orgId=1&refresh=10s>

JVM-метрики, формируемые приложением Датастюард:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Наименование | Тип | Описание | Метки | Комментарий |
| 1 | jvm\_memory\_committed\_bytes | gauge | Объем памяти в байтах, выделенный для JVM | 1. area:
	1. nonheap
	2. heap
2. id
 |  |
| 2 | jvm\_threads\_states\_threads | gauge | Текущее количество потоков | 1. state:
	1. runnable
	2. blocked
	3. waiting
	4. timed-waiting
	5. new
	6. terminated
 |  |
| 3 | jvm\_buffer\_memory\_used\_bytes | gauge | Оценка объема памяти, который JVM использует для этого буферного пула | id |  |
| 4 | jvm\_memory\_used\_bytes | gauge | Объем используемой памяти | 1. area:
	1. nonheap
	2. heap
2. id
 |  |
| 5 | jvm\_gc\_live\_data\_size\_bytes | gauge | Размер пула долговременной памяти кучи после восстановления | - |  |
| 6 | jvm\_memory\_max\_bytes | gauge | Максимальный объем памяти в байтах, который может быть использован для управления памятью | 1. area:
	1. nonheap
	2. heap
2. id
 |  |
| 7 | jvm\_buffer\_count\_buffers | gauge | Оценка количества буферов в пуле | id |  |
| 8 | jvm\_gc\_memory\_allocated\_bytes\_total | counter | Увеличение для роста размера (нового) пула памяти кучи после одного GC (garbage collection process) до следующего | - | Исходное описание: Incremented for an increase in the size of the (young) heap memory pool after one GC to before the next |
| 9 | jvm\_threads\_peak\_threads | gauge | Максимальное количество активных потоков с момента запуска JVM или сброса пика | - |  |
| 10 | jvm\_gc\_max\_data\_size\_bytes | gauge | Максимальный размер пула долговременной памяти кучи | - |  |
| 11 | jvm\_classes\_loaded\_classes | gauge | Количество классов, которые в данный момент загружены в JVM | - |  |
| 12 | jvm\_classes\_unloaded\_classes\_total | counter | Общее количество классов, выгруженных с момента запуска JVM | - |  |
| 13 | jvm\_gc\_pause\_seconds | summary | Время, затраченное на паузу GC | 1. action
2. cause
 |  |
| 14 | jvm\_gc\_pause\_seconds\_max | gauge | Максимальное время, затраченное на паузу GC | 1. action
2. cause
 |  |
| 15 | jvm\_buffer\_total\_capacity\_bytes | gauge | Оценка общей емкости буферов в этом пуле | id |  |
| 16 | jvm\_gc\_memory\_promoted\_bytes\_total | counter | Число увеличений размера ранее выделенной памяти до GC и после GC | - | Исходное описание: Count of positive increases in the size of the old generation memory pool before GC to after GC |
| 17 | jvm\_threads\_live\_threads | gauge | Текущее число активных потоков (включая daemon и non-daemon потоки) | - |  |
| 18 | jvm\_threads\_daemon\_threads | gauge | Текущее количество активных потоков (daemon) |  |  |

Прикладные метрики, формируемые приложением Датастюард:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Метрика | Тип | Описание |
| 1 | req\_count\_total(result = success) | Counter | Количество обработанных запросов с указанием типа запроса в kind. Типы запросов:* statistics – запрос получения статистики витрины
* qualityResultRequest – запрос получения результатов проверок качества
* qualityListRequest – запрос получения справочника проверок качества
* dataRequest – запрос получения произвольных данных витрины
 |
| 2 | req\_count\_total(result = failure) | Counter | Количество ответов с ошибками с указанием типа запроса в kind. Типы ответов:* statistics – ошибка получения статистики
* qualityResultRequest – ошибка получения результатов проверок качества
* qualityListRequest – ошибка получения справочника проверок качества
* dataRequest – ошибка получения произвольных данных витрины
 |
| 3 | req\_time\_seconds | Histogram | Время обработки входящего запроса с указанием типа в kind. Типы запросов:* statistics – запрос получения статистики витрины
* qualityResultRequest – запрос получения результатов проверок качества
* qualityListRequest – запрос получения справочника проверок качества
* dataRequest – запрос получения произвольных данных витрины
 |

# Приложение Б. Описание спецификации для взаимодействия интерфейсов ТПО УД

## Датамарты

### Получение данных для выбора датамарта

GET/ api/v1/datamarts

Запрос на получение списка датамартов, отправляется без параметров.

URL: <http://localhost/api/v1/datamarts>

Пример запроса:

curl --location 'http://localhost/api/v1/datamarts' \

--header 'Accept: application/json'

Пример ответа:

 "datamarts": [

 "datamart\_001",

 "datamart\_002",

 "datamart\_003"

 ],

 "default": "datamart\_001"

}

### Получение данных таблиц датамарта

GET/ api/v1/datamarts/{datamart}/tables

Запрос на получение данных для выпадающего списка таблиц в фильтре экрана «Ошибки в данных».

URL: [http://localhost/api/v1/datamarts/:datamart/tables](http://localhost/api/v1/datamarts/%3Adatamart/tables)

Пример запроса:

curl --location 'http://localhost/api/v1/datamarts/<string>/tables' \

--header 'Accept: application/json'

Параметры запроса

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Тип запроса** | **Обязательность** | **Описание** | **Пример** |
| datamart | string | да | мнемоника датамарта | mnemonic\_01 |

Пример ответа:

{

 "tables": [

 "students",

 "books",

 "organiztion"

 ]

}

### Получение данных справочника проверок

GET/ api/v1/datamarts/{datamart}/quality-checks

Запрос на получение данных справочника проверок.

URL: [http://localhost/api/v1/datamarts/:datamart/quality-checks?refresh=<boolean](http://localhost/api/v1/datamarts/%3Adatamart/quality-checks?refresh=%3cboolean)>

Пример запроса:

curl --location 'http://localhost/api/v1/datamarts/<string>/quality-checks?refresh=%3Cboolean%3E' \

--header 'Accept: application/json'

Параметры запроса

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Тип запроса** | **Обязательность** | **Описание** | **Пример** |
| datamart | string | да | мнемоника датамарта | mnemonic\_01 |
| refresh | boolean | нет | указание необходимости обновления кэша проверок | true |

Пример ответа:

{

 "checks": [

 {

 "mnemonic": "mnemonic\_qc2",

 "name": "Проверка полноты данных таблицы книг",

 "table": "books",

 "id": "111",

 "kind": "Блокирующая (пороговый инцидент)",

 "type": "Полнота"

 },

 {

 "mnemonic": "mnemonic\_qc1",

 "name": "Проверка формата СНИЛС у студентов",

 "table": "students",

 "id": "110",

 "kind": "Блокирующая (построчный инцидент)",

 "type": "Достоверность"

 }

 ]

}

### Получение результатов проверок (Ошибки в данных)

GET/ api/v1/datamarts/{datamart}/tables/{table}/quality-results

Запрос на получение данных таблицы с результатами проверок.

URL: [http://localhost/api/v1/datamarts/:datamart/tables/:table/quality-results?dqf-algorithm-id=<string>&start-at=<string>&finish-at=<string>&dqf-id=<string>&dqf-kind=<string>&page=1&size=100](http://localhost/api/v1/datamarts/%3Adatamart/tables/%3Atable/quality-results?dqf-algorithm-id=%3cstring%3e&start-at=%3cstring%3e&finish-at=%3cstring%3e&dqf-id=%3cstring%3e&dqf-kind=%3cstring%3e&page=1&size=100)

Пример запроса:

curl --location 'http://localhost/api/v1/datamarts/<string>/tables/<string>/quality-results?dqf-algorithm-id=%3Cstring%3E&start-at=%3Cstring%3E&finish-at=%3Cstring%3E&dqf-id=%3Cstring%3E&dqf-kind=%3Cstring%3E&page=1&size=100' \

--header 'Accept: application/json'

Параметры запроса

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Тип запроса** | **Обязательность** | **Описание** | **Пример** |
| datamart | string | да | мнемоника датамарта | mnemonic\_01 |
| table | string | да | наименование таблицы | students |
| dqf-algorithm-id | string | нет | идентификатор(ы) проверки | 110, 111 |
| start-at | timestamp | нет | дата и время начала периода проверок | 1727859181 |
| finish-at | timestamp | нет | дата и время окончания периода проверок | 1727859181 |
| dqf-id | string | нет | идентификатор записи таблицы errors | 1ece0a89-039e-4dad-9a1f-e99d9a322705 |
| dqf-kind | string | нет | вид проверки | 2, 3 |
| page  | string | да | номер страницы | 1 |
| size  | string | да | количество результатов на странице | 1 |

Пример ответа:

{

 "data": [

 {

 "dqf-id": "163ffac9-b27a-4a61-8775-b80e81eb8c92",

 "pk-fields": [

 {

 "name": "pkey\_1",

 "value": "30325990-7edb-4b81-a6fb-d901887ab831"

 },

 {

 "name": "pkey\_2",

 "value": "30325990-7edb-4b81-a6fb-d901887ab832"

 }

 ],

 "dqf-table": "students",

 "dqf-algorithm-id": 110,

 "dqf-code": "Некорректный СНИЛС",

 "dqf-description": "status\_code: active, status\_value: -1",

 "dqf-checked-at": "1727859181",

 "dqf-version": 8,

 "mnemonic": "mnemonic\_qc2"

 }

 ],

 "pagination": {

 "page": 1,

 "size": 100,

 "total": 577

 }

}

### Получение детальной информации о записи основной таблицы

GET/ api/v1/datamarts/{datamart}/tables/{table}/quality-results/{dqf-id}

Запрос на получение данных о записи основной таблицы.

URL: [http://localhost/api/v1/datamarts/:datamart/tables/:table/quality-results/:dqf-id](http://localhost/api/v1/datamarts/%3Adatamart/tables/%3Atable/quality-results/%3Adqf-id)

Пример запроса:

curl --location 'http://localhost/api/v1/datamarts/<string>/tables/<string>/quality-results/<string>' \

--header 'Accept: application/json'

Параметры запроса

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Тип запроса** | **Обязательность** | **Описание** | **Пример** |
| datamart | string | да | мнемоника датамарта | mnemonic\_01 |
| table | string | да | наименование таблицы | students |
| dqf-id | string | да | идентификатор записи ошибки | d61246c9-fe6a-407e-9e8b-881a25248601 |

Пример ответа:

{

 "quality-result": {

 "dqf-id": "163ffac9-b27a-4a61-8775-b80e81eb8c92",

 "pk-fields": [

 {

 "name": "pkey\_1",

 "value": "30325990-7edb-4b81-a6fb-d901887ab831"

 },

 {

 "name": "pkey\_2",

 "value": "30325990-7edb-4b81-a6fb-d901887ab832"

 }

 ],

 "dqf-table": "students",

 "dqf-algorithm-id": 110,

 "dqf-code": "Некорректный СНИЛС",

 "dqf-description": "status\_code: active, status\_value: -1",

 "dqf-checked-at": "1727859181",

 "dqf-version": 8,

 "mnemonic": "mnemonic\_qc2"

 },

 "data": [

 {

 "field-name": "pkey\_1",

 "value": "2d0bddf5-08ad-4974-b176-c0c1560fd374",

 "error": false

 },

 {

 "field-name": "LastName",

 "value": "!900434ff;",

 "error": true

 },

 {

 "field-name": "FirstName",

 "value": "Энакин",

 "error": false

 }

 ]

}

### Выгрузка результатов проверок (Ошибки в данных) в формате CSV

GET/ api/v1/datamarts/{datamart}/tables/{table}/csv

Запрос на выгрузку данных таблицы с результатами проверок в формате CSV.

URL: [http://localhost/api/v1/datamarts/:datamart/tables/:table/csv?dqf-algorithm-id=<string>&start-at=<string>&finish-at=<string>&dqf-id=<string>&dqf-kind=<string](http://localhost/api/v1/datamarts/%3Adatamart/tables/%3Atable/csv?dqf-algorithm-id=%3cstring%3e&start-at=%3cstring%3e&finish-at=%3cstring%3e&dqf-id=%3cstring%3e&dqf-kind=%3cstring)>

Пример запроса:

curl --location 'http://localhost/api/v1/datamarts/<string>/tables/<string>/csv?dqf-algorithm-id=%3Cstring%3E&start-at=%3Cstring%3E&finish-at=%3Cstring%3E&dqf-id=%3Cstring%3E&dqf-kind=%3Cstring%3E' \

--header 'Accept: text/csv'

Параметры запроса

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Тип запроса** | **Обязательность** | **Описание** | **Пример** |
| datamart | string | да | мнемоника датамарта | mnemonic\_01 |
| table | string | да | наименование таблицы | students |
| dqf-algorithm-id | string | нет | идентификатор(ы) проверки | 110, 111 |
| start-at | timestamp | нет | дата и время начала периода проверок | 1727859181 |
| finish-at | timestamp | нет | дата и время окончания периода проверок | 1727859181 |
| dqf-id | string | нет | идентификатор записи таблицы errors | 1ece0a89-039e-4dad-9a1f-e99d9a322705 |
| dqf-kind | string | нет | вид проверки | 2, 3 |

Пример ответа:

dqf-id,pk-fields,dqf-table,dqf-algorithm-id,dqf-code,dqf-description,dqf-checked-at,dqf-version

"550e8400-e29b-41d4-a716-446655440000","name:value1,name:value2","students","110","Некорректный СНИЛС","snils:111-223!#","2025-02-05 06:58:03","8"

"f47ac10b-58cc-4372-a567-0e02b2c3d479","name:value3,name:value4","students","111","Не заполнена фамилия","lastName:","2025-02-05 06:58:03","15"

### Получение детальной информации о проверке

GET/ api/v1/datamarts/{datamart}/quality-checks/{qualityCheckId}

Запрос на получение детальной информации о проверке.

URL: [http://localhost/api/v1/datamarts/:datamart/quality-checks/:qualityCheckId](http://localhost/api/v1/datamarts/%3Adatamart/quality-checks/%3AqualityCheckId)

Пример запроса:

curl --location 'http://localhost/api/v1/datamarts/<string>/quality-checks/<integer>' \

--header 'Accept: application/json'

Параметры запроса

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Тип запроса** | **Обязательность** | **Описание** | **Пример** |
| datamart | string | да | мнемоника датамарта | mnemonic\_01 |
| qualityCheckId | integer | да | идентификатор записи проверки | 110 |

Пример ответа:

{

 "data": {

 "mnemonic": "mnemonic\_qc2",

 "name": "Проверка формата СНИЛС у студентов",

 "description": "Проверка формата СНИЛС записей таблицы students по заданной маске шаблона",

 "table": "students",

 "id": 110,

 "version-code": "1.1",

 "kind": "Уведомительная",

 "type": "Достоверность",

 "yaml-schema": "CheckValue: $.objects.fields.atr\_1 != null otherwise: - AppendValue: expr: code: \"atr\_1...",

 "error-rate": 1,

 "dqf-check-external-source": "ЕРН",

 "start-type": "По загрузке данных",

 "datamart-attributes": [

 {

 "id": 228800,

 "name": "СНИЛС",

 "tech-name": "snils"

 }

 ],

 "period": {

 "periodicity": "Ежедневно",

 "time": "12:00:00",

 "days-of-week": "Понедельник, Среда, Пятница",

 "day-of-month": 5,

 "period-manual-interval": "2024-10-08 - 2024-12-08",

 "period-from": "2024-10-08",

 "period-to": "2024-12-08",

 "created-at": "2024-11-02T16:32:13.022+03:00"

 },

 "pebble-schema": {

 "pebble-request": "{% set saved\_cn\_fl = sdbr (\"cn\_fl\") | default (\"-1\") | toLong() %}...",

 "pebble-response": "{% var algoritm %} 123 {% endvar %} {{ log(\"error\", \"Получен ответ от ФНС с данными об ошибках ФЛ \") }}...",

 "namespace-uri": "urn://x-artefacts-fns-idern/root/1.0.0",

 "response-root-tag": "IDERNResponse"

 }

 }

}

### Получение статистики по выполненным проверкам

GET/ api/v1/datamarts/{datamart}/statistics/completed-checks

Запрос на получение статистики по выполненным проверкам качества.

URL: [http://localhost/api/v1/datamarts/:datamart/statistics/completed-checks?page=1&size=100](http://localhost/api/v1/datamarts/%3Adatamart/statistics/completed-checks?page=1&size=100)

Пример запроса:

curl --location 'http://localhost/api/v1/datamarts/<string>/statistics/completed-checks?page=1&size=100' \

--header 'Accept: application/json'

Параметры запроса

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Тип запроса** | **Обязательность** | **Описание** | **Пример** |
| datamart | string | да | мнемоника датамарта | mnemonic\_01 |

Пример ответа:

{

 "data": {

 "completed-checks": [

 {

 "algorithm-id": "110",

 "table": "books",

 "checked-at": "1727859181",

 "field-count": "1000",

 "errors-count": "150",

 "perc-quality": "85%",

 "id": "73158b55-f767-404b-8a18-7cdc5440999d",

 "mnemonic": "mnemonic\_qc2",

 "type": "Достоверность"

 },

 {

 "algorithm-id": "111",

 "table": "students",

 "checked-at": "1727859181",

 "field-count": "1000",

 "errors-count": "100",

 "perc-quality": "90%",

 "id": "82112460-db89-4755-a8a4-d7e8347acbe0",

 "mnemonic": "mnemonic\_qc3",

 "type": "Полнота"

 }

 ]

 },

 "pagination": {

 "page": 1,

 "size": 100,

 "total": 577

 }

}

### Получение статистики по инцидентам

GET/ api/v1/datamarts/{datamart}/statistics/incidents

Запрос на получение агрегированной статистики по инцидентам.

URL: [http://localhost/api/v1/datamarts/:datamart/statistics/incidents](http://localhost/api/v1/datamarts/%3Adatamart/statistics/incidents)

Пример запроса:

curl --location 'http://localhost/api/v1/datamarts/<string>/statistics/incidents' \

--header 'Accept: application/json'

Параметры запроса

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Тип запроса** | **Обязательность** | **Описание** | **Пример** |
| datamart | string | да | мнемоника датамарта | mnemonic\_01 |

Пример ответа:

{

 "data": {

 "total-incidents": 3,

 "total-incidents-perc": "100%",

 "expired-incidents": 0,

 "expired-incidents-perc": "0%",

 "not-expired-incidents": 3,

 "not-expired-incidents-perc": "100%"

 }

}

### Получение статистики по строкам и ошибкам

GET/ api/v1/datamarts/{datamart}/statistics/errors

Запрос на получение агрегированной статистики по строкам и ошибкам.

URL: [http://localhost/api/v1/datamarts/:datamart/statistics/errors](http://localhost/api/v1/datamarts/%3Adatamart/statistics/errors)

Пример запроса:

curl --location 'http://localhost/api/v1/datamarts/<string>/statistics/errors' \

--header 'Accept: application/json'

Параметры запроса

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Тип запроса** | **Обязательность** | **Описание** | **Пример** |
| datamart | string | да | мнемоника датамарта | mnemonic\_01 |

Пример ответа:

{

 "data": {

 "update-ts": "1727859181",

 "data-indicators": {

 "total-rows": 10,

 "total-rows-perc": "100%",

 "block-rows": 2,

 "block-rows-perc": "20%",

 "error-rows": 5,

 "error-rows-perc": "50%"

 },

 "errors-indicators": {

 "total-errors": 6,

 "total-errors-perc": "100%",

 "block-errors": 3,

 "block-errors-perc": "50%",

 "notice-errors": 3,

 "notice-errors-perc": "50%"

 }

 }

}

### Получение статистики ошибок по таблицам

GET/ api/v1/datamarts/{datamart}/statistics/tables

Запрос на получение агрегированной статистики по таблицам.

URL: [http://localhost/api/v1/datamarts/:datamart/statistics/tables?table=<string>&page=1&size=100](http://localhost/api/v1/datamarts/%3Adatamart/statistics/tables?table=%3cstring%3e&page=1&size=100)

Пример запроса:

curl --location 'http://localhost/api/v1/datamarts/<string>/statistics/tables?table=%3Cstring%3E&page=1&size=100' \

--header 'Accept: application/json'

Параметры запроса

| **Наименование** | **Тип запроса** | **Обязательность** | **Описание** | **Пример** |
| --- | --- | --- | --- | --- |
| datamart | string | да | мнемоника датамарта | mnemonic\_01 |
| table | string | да | наименование таблицы | students |
| page  | string | да | номер страницы | 1 |
| size  | string | да | количество результатов на странице | 1 |

Пример ответа:

{

 "data": {

 "tables-errors": [

 [

 {

 "table": "students",

 "block-errors": 3,

 "total-errors": 6

 },

 {

 "table": "books",

 "block-errors": 15,

 "total-errors": 526

 }

 ]

 ]

 },

 "pagination": {

 "page": 1,

 "size": 100,

 "total": 577

 }

}

### Произвольный SQL-запрос

POST/ api/v1/datamarts/{datamart}/sql

Произвольный запрос на получение данных витрины.

URL: [http://localhost/api/v1/datamarts/:datamart/sql?page=1&size=100](http://localhost/api/v1/datamarts/%3Adatamart/sql?page=1&size=100)

Пример запроса:

curl --location 'http://localhost/api/v1/datamarts/<string>/sql?page=1&size=100' \

--header 'Content-Type: application/json' \

--header 'Accept: application/json' \

--data '<string>'

Параметры запроса

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Тип запроса** | **Обязательность** | **Описание** | **Пример** |
| datamart | string | да | мнемоника датамарта | mnemonic\_01 |
| page  | string | да | номер страницы | 1 |
| size  | string | да | количество результатов на странице | 1 |

Пример ответа:

{

 "data": {

 "fields": [

 "id",

 "org",

 "lastName"

 ],

 "lines": [

 [

 {

 "field": "id",

 "value": "f3efd928-e1c0-47c6-bf96-fc368d37468b"

 },

 {

 "field": "org",

 "value": "МИСиС"

 },

 {

 "field": "lastName",

 "value": "Иванов"

 }

 ]

 ]

 },

 "pagination": {

 "page": 1,

 "size": 100,

 "total": 577

 },

 "error": {

 "code": "FORBIDDEN\_SQL",

 "message": "Запрещенный синтаксис в sql-выражении"

 }

}

## Инциденты

### Получение данных об инциденте

GET/ api/v1/incidents/{incident}

Запрос на получение данных об инциденте.

URL: [http://localhost/api/v1/incidents/:incident](http://localhost/api/v1/incidents/%3Aincident)

Пример запроса:

curl --location 'http://localhost/api/v1/incidents/<string>' \

--header 'Accept: application/json'

Параметры запроса

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Тип запроса** | **Обязательность** | **Описание** | **Пример** |
| incident | string | да | номер инцидента | ЕИП-0001 |

Пример ответа:

{

 "data": {

 "table": "books",

 "dqf\_id": "1ece0a89-039e-4dad-9a1f-e99d9a322705",

 "dqf\_algorithm\_id": "students\_snils"

 },

 "error": {

 "code": "NOT\_FOUND",

 "message": "Инцидент не найден"

 }

}

### Получение списка инцидентов

GET/ api/v1/incidents/{datamart}/list

Запрос на получение списка инцидентов.

URL: [http://localhost/api/v1/incidents/:datamart/list?number=<string>&expired=<boolean>&page=1&size=100](http://localhost/api/v1/incidents/%3Adatamart/list?number=%3cstring%3e&expired=%3cboolean%3e&page=1&size=100)

Пример запроса:

curl --location 'http://localhost/api/v1/incidents/<string>/list?number=%3Cstring%3E&expired=%3Cboolean%3E&page=1&size=100' \

--header 'Accept: application/json'

Параметры запроса

| **Наименование** | **Тип запроса** | **Обязательность** | **Описание** | **Пример** |
| --- | --- | --- | --- | --- |
| datamart | string | да | мнемоника датамарта | mnemonic\_01 |
| number | string | нет | номер инцидента | 000111 |
| expired | boolean | нет | Признак, указывающий просрочен инцидент или нет | true |
| page  | string | да | номер страницы | 1 |
| size  | string | да | количество результатов на странице | 1 |

Пример ответа:

{

 "data": {

 "incidents": [

 {

 "number": "000111",

 "created-at": "1727859181",

 "expired": false,

 "check-kind": "Блокирующая (построчный инцидент)",

 "check-mnemonic": "Проверка формата СНИЛС",

 "table": "students"

 }

 ]

 },

 "pagination": {

 "page": 1,

 "size": 100,

 "total": 577

 }

}

## Пользователи

### Получение данных о пользователе

GET/ api/v1/users/current

Запрос на получение данных об авторизованном пользователе.

URL: <http://localhost/api/v1/users/current>

Пример запроса:

curl --location 'http://localhost/api/v1/users/current' \

--header 'Accept: application/json'

Пример ответа:

{

 "sub": "62873e21-16c9-4fb6-a067-81cc2619600a",

 "email\_verified": false,

 "gender": "male",

 "name": "Иван",

 "preferred\_username": "ivanovii",

 "middle\_name": "Иванович",

 "given\_name": "Иван",

 "family\_name": "Иванов",

 "email": "ivanovii@example.ru"

}

### Выход пользователя из ТПО УД

POST/ api/v1/users/current/logout

Запрос для выполнения logout пользователя.

URL: <http://localhost/api/v1/users/current/logout>

Пример запроса:

curl --location --request POST 'http://localhost/api/v1/users/current/logout' \

--header 'Accept: application/json'

Пример ответа:

{

 "status": "ОК"

}

## Ошибки

Ошибки выводятся в блоке Responses

|  |  |  |
| --- | --- | --- |
| **Код ошибки** | **Описание** | **Пример** |
| 400 | Bad Request Ошибка парсинга сообщения | { "error": { "code": "BAD\_REQUEST", "message": "Не передан обязательный параметр <datamart>" }} |
| 500 | Internal Server Error Внутренняя ошибка | { "error": { "code": "INTERNAL\_ERROR", "message": "IllegalArgumentException: <подробности ошибки, опционально>, errorId: <идентификатор ошибки>" }} |

# Термины и определения

| Термин | Определение |
| --- | --- |
| Типовое программное обеспечение управления данными(ТПО УД) | Механизм контроля качества данных, размещённых на программном обеспечении «Витрина данных», через Единую систему межведомственного электронного взаимодействия версии 3.х |
| Витрина (данных) | Комплекс программных и технических средств в составе информационно-телекоммуникационной инфраструктуры участника взаимодействия, обеспечивающий хранение и предоставление данных другим участникам взаимодействия в соответствии с критериями, установленными Министерством цифрового развития, связи и массовых коммуникаций |
| Гб | Гигабайт |
| ЕИП НСУД | Единая информационная платформа национальной системы управления данными |
| ОС | Операционная система |
| Ответчик | Участник взаимодействия, на стороне которого развернут REST-сервис ИС Ответчика и который является получателем запроса |
| Регламентированный REST-запрос (Запрос к REST-сервису ИС Ответчика) | Регламентированный запрос, представляющий из себя REST-сервис в инфраструктуре Ответчика, обеспечивающий предоставление данных другим Участникам взаимодействия с использованием СМЭВ4 в соответствии с загруженной в СМЭВ4 спецификацией OpenAPI |
| Регламентированный SQL-запрос (РЗ) | SQL-запрос, выраженный в терминах Модели данных, загруженной в СМЭВ4, и зарегистрированный в Ядре СМЭВ4 под символической мнемоникой, используемой ИС Потребителя СМЭВ4 для выполнения регламентированного запроса. Может иметь параметры, значения которых задаются Потребителем данных СМЭВ4 при выполнении регламентированного запроса |
| СМЭВ4 | Единый сервис доступа к данным СМЭВ, предназначенный для автоматизации процесса передачи данных и уведомлений об изменении данных между организациями или органами власти, ответственными за формирование и ведение информационных ресурсов, зарегистрированных в ЕИП НСУД |
| УВ | Участник взаимодействия – федеральный орган исполнительной власти, государственный внебюджетный фонд и иной орган или организация, участвующие в предоставлении государственных и муниципальных услуг (функций) |
| ЦПУ | Центральное процессорное устройство |
| API | (англ. Application Programming Interface) — набор классов, процедур, функций, структур или констант, которыми одна компьютерная программа может взаимодействовать с другой программой |
| OpenAPI, спецификация OpenAPI | Формализованная спецификация и экосистема множества инструментов, предоставляющая интерфейс между front-end системами, кодом библиотек низкого уровня и коммерческими решениями в виде API |
| REST | (англ. Representational State Transfer) — Архитектурный стиль взаимодействия компонентов распределённого приложения в сети |
| REST-сервис ИС Ответчика | Сервис в инфраструктуре Ответчика, зарегистрированный в СМЭВ4 и обеспечивающий обмен данными с другими Участниками взаимодействия с использованием СМЭВ4 в соответствии с загруженной в СМЭВ4 спецификацией OpenAPI |
| SQL | (англ. Structured Query Language) — декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных |

# История изменений документа

Таблица 4 - История изменений документа

|  |  |
| --- | --- |
| Версия | Изменения документа |
| 1.0 | Документ создан |
| 1.0.1 | Обновлен раздел:* 2.2.3 Перечень стороннего ПО, необходимого для работы ТПО УД;
* 4.3.2 Настройка параметров подключения к Ядру;
* 4.3.4 Настройка параметров Витрин;
* 4.3.6 Настройка ТПО УД для передачи метрик в Prometheus;

Добавлены разделы:* 4.3.3 Настройка использования альтернативного значения серверного порта
* 4.3.7 Настройка количества одновременных соединений;
* 4.3.8 Настройка максимального количества строк в CSV файле;
* Приложение А. Описание метрик;
* Приложение Б. Описание спецификации для взаимодействия интерфейсов ТПО УД.
 |

1. Данные порты могут быть переопределены посредством конфигурационного файла [↑](#footnote-ref-1)