**Методические рекомендации по работе   
с подсистемой обеспечения доступа к данным федеральной государственной информационной системы «Единая система межведомственного электронного взаимодействия»  
Версия 2.10.0.0**

**ПОДД СМЭВ**

Листов 109

Москва 2022

Содержание

[1. Введение 4](#_Toc112405407)

[1.1. Общие сведения 4](#_Toc112405408)

[**1.1.1.** **Номер версии документа** 4](#_Toc112405409)

[1.1.2. Нормативно-правовые основания 4](#_Toc112405410)

[1.2. Виды информационного обмена с использованием СМЭВ 5](#_Toc112405411)

[1.2.1. Классификация регламентированных запросов 6](#_Toc112405412)

[1.3. Участники информационного обмена с использованием ПОДД СМЭВ 7](#_Toc112405413)

[1.4. Информационный обмен участников взаимодействия с использованием ПОДД СМЭВ 8](#_Toc112405414)

[1.4.1. Обмен с использованием SQL-запросов 8](#_Toc112405415)

[1.4.2. Обмен с использованием Подписок на уведомления об изменениях 12](#_Toc112405416)

[1.4.3. Обмен с использованием запросов к REST-сервису ИС Поставщика 18](#_Toc112405417)

[1.5. Требования к участникам взаимодействия для использования ПОДД СМЭВ 23](#_Toc112405418)

[1.5.1. Требования к Поставщикам данных 23](#_Toc112405419)

[1.5.2. Требования к Потребителям данных 25](#_Toc112405420)

[1.6. Метаданные ПОДД СМЭВ 25](#_Toc112405421)

[1.6.1. Модель данных Витрины Поставщика данных 28](#_Toc112405422)

[1.6.2. Подписки Потребителей данных ПОДД СМЭВ 29](#_Toc112405423)

[1.6.3. Определения Регламентированных запросов 30](#_Toc112405424)

[1.6.4. Спецификации OpenAPI для REST-сервиса ИС Поставщика 31](#_Toc112405425)

[1.6.5. Полномочия Потребителя данных ПОДД СМЭВ на доступ к данным 36](#_Toc112405426)

[1.7. Типы данных ПОДД СМЭВ 38](#_Toc112405427)

[2. Описание подключения к ПОДД СМЭВ 41](#_Toc112405428)

[2.1. Протокол ПОДД СМЭВ 41](#_Toc112405429)

[2.2. Подключение участников взаимодействия с использованием ПОДД СМЭВ 41](#_Toc112405430)

[2.3. Протокол взаимодействия Агента ПОДД СМЭВ и Витрины Поставщика данных 41](#_Toc112405431)

[2.3.1. Перечень заголовков сообщений в Apache Kafka 41](#_Toc112405432)

[2.3.2. Перечень топиков брокера сообщений Apache Kafka 42](#_Toc112405433)

[2.3.3. Настройка Агента ПОДД для работы с несколькими Витринами данных 44](#_Toc112405434)

[2.3.4. Последовательность исполнения SQL-запроса в контуре Поставщика данных 47](#_Toc112405435)

[2.3.5. Структуры сообщений для взаимодействия с Поставщиком 47](#_Toc112405436)

[2.3.6. Примеры реализации взаимодействия с Агентом ПОДД СМЭВ с использованием брокера сообщений Apache Kafka 63](#_Toc112405437)

[2.4. Протокол взаимодействия Агента ПОДД СМЭВ и ИС Потребителя данных 68](#_Toc112405438)

[2.4.1. REST-интерфейс Агента ПОДД СМЭВ 68](#_Toc112405439)

[2.4.2. JDBC-интерфейс Агента ПОДД СМЭВ 77](#_Toc112405440)

[2.5. Протокол взаимодействия Агента ПОДД СМЭВ и Хранилища данных по подписке Потребителя данных 79](#_Toc112405441)

[2.5.1. Перечень топиков брокера сообщений Apache Kafka 80](#_Toc112405442)

[2.5.2. Структуры сообщений для взаимодействия с Потребителем данных 80](#_Toc112405443)

[2.6. Протокол взаимодействия Агента ПОДД СМЭВ и ИС Поставщика данных 83](#_Toc112405444)

[3. Использование ПОДД СМЭВ 85](#_Toc112405445)

[3.1. SQL-синтаксис 85](#_Toc112405446)

[3.1.1. Поддерживаемые возможности SQL-синтаксиса 87](#_Toc112405447)

[3.2. Примеры запросов с использованием SQL-синтаксиса, поддерживаемого ПОДД СМЭВ 91](#_Toc112405448)

[3.2.1. Запрос с использованием табличных выражений 92](#_Toc112405449)

[3.2.2. Запросы с использованием базовых операторов 92](#_Toc112405450)

[3.2.3. Запросы с указанием версии Витрины 92](#_Toc112405451)

[3.2.4. Запросы с использованием JOIN 92](#_Toc112405452)

[3.2.5. Регламентированные запросы 93](#_Toc112405453)

[3.3. Получение двоичных объектов в результатах запроса 93](#_Toc112405454)

[3.4. Формирование и передача статистики атрибутов Витрины данных Агенту ПОДД СМЭВ 95](#_Toc112405455)

[3.5. Выполнение регламентированных запросов 96](#_Toc112405456)

[3.6. Выполнение запросов с использованием табличных параметров, передаваемых Потребителем данных для обогащения 97](#_Toc112405457)

[3.6.1. Запрос с использованием REST-интерфейса 97](#_Toc112405458)

[3.6.2. Запрос с использованием JDBC-интерфейса 97](#_Toc112405459)

[3.7. Выполнение запросов к REST-сервису ИС Поставщика 98](#_Toc112405460)

[Приложение 1. Термины и сокращения 99](#_Toc112405461)

[Приложение 2. Пример проекта для реализации взаимодействия с Агентом ПОДД СМЭВ 109](#_Toc112405462)

# Введение

## Общие сведения

Настоящий документ описывает правила и приёмы использования возможностей инфраструктуры электронного правительства (далее – ИЭП), которые позволяют участникам межведомственного взаимодействия (далее – УВ) решать задачи передачи сведений с использованием подсистемы обеспечения доступа к данным (далее – ПОДД) системы межведомственного электронного взаимодействия (далее – СМЭВ) между информационными системами участников взаимодействия (далее – ИС УВ).

Термины и сокращения, используемые в данном документе, представлены в Приложении 1.

* + 1. **Номер версии документа**

Номер версии документа формируется по шаблону A.X.Y.Z, где:

A – номер версии подсистемы обеспечения доступа к данным федеральной государственной информационной системы «Единая система межведомственного электронного взаимодействия».

X – номер поколения документа. Изменение данного номера означает значительные изменения в структуре и содержании документа.

Y – номер основного релиза документа в рамках поколения. Документ может содержать освещение новых и/или незначительную переработку содержащихся в предыдущей версии документа тем. Плановая подготовка основного релиза документа осуществляется раз в квартал. Основные релизы утверждаются Подкомиссией по использованию информационных технологий при предоставлении государственных и муниципальных услуг Правительственной комиссии по внедрению информационных технологий в деятельность государственных органов и органов местного самоуправления.

Z – номер технологического релиза в рамках основного релиза. Может содержать в себе стилистические, редакционные, незначительные технические изменения. Данный тип релизов выпускается по необходимости и не проходит специализированной процедуры утверждения Подкомиссией по использованию информационных технологий при предоставлении государственных и муниципальных услуг Правительственной комиссии по внедрению информационных технологий в деятельность государственных органов и органов местного самоуправления.

### Нормативно-правовые основания

Данный документ разработан в целях реализации и во исполнение:

* Постановления Правительства Российской Федерации от 8 сентября 2010 года № 697 «О единой системе межведомственного электронного взаимодействия»;
* Распоряжения Правительства Российской Федерации от 3 июня 2019 года №1189-р «Об утверждении Концепции создания и функционирования национальной системы управления данными и плана мероприятий («дорожную карту») по созданию национальной системы управления данными на 2019–2021 годы» (далее – Концепция);
* Постановления Правительства Российской Федерации от 3 июня 2019 года № 710 «О проведении эксперимента по повышению качества и связанности данных, содержащихся в государственных информационных ресурсах»;
* Федерального закона от 27 июля 2010 г. № 210-ФЗ «Об организации предоставления государственных и муниципальных услуг»;
* мероприятий федерального проекта «Цифровое государственное управление» национальной программы «Цифровая экономика», утверждённой на заседании правительственной комиссии по цифровому развитию, использованию информационных технологий для улучшения качества жизни и условий ведения предпринимательской деятельности под председательством Председателя Правительства Российской Федерации Д. А. Медведева. 25 декабря 2018 года:
  + 06.01.011.001.004 «Разработка функциональных и технических требований к информационным системам НСУД (включая требования к функциям получения, очистки и преобразования данных, хранения и обработки данных, визуального представления данных, управления метаданными, межведомственного взаимодействия);
  + 06.01.011.001.006 Разработка, адаптация программного обеспечения и разработка архитектуры и проектных решений на НСУД и ее части, внедрение функционала НСУД, включая пусконаладочные работы, проведение предварительных испытаний, проведение опытной эксплуатации, доработку программного обеспечения, дополнительную наладку технических средств и проведение приемочных испытаний НСУД в соответствии с 06.01.011.001.005.

## Виды информационного обмена с использованием СМЭВ

Виды информационного обмена, поддерживаемые СМЭВ, приведены в таблице Таблица 1. Данный документ описывает рекомендации и требования в отношении 2–4 видов обмена.

Таблица 1 – Виды информационного обмена

| **№** | **Вид информационного обмена** | **Характеристика** |
| --- | --- | --- |
| 1 | Обмен с использованием Видов Сведений | Обмен с фиксированной структурой запроса и ответа с использованием особого протокола обмена – Вид Сведения (далее – ВС).  Описание данного вида обмена приведено в документе «Методические рекомендации по работе с Единой системой межведомственного электронного взаимодействия», размещенном на технологическом портале СМЭВ 3, <https://smev3.gosuslugi.ru/portal/> |
| 2 | Обмен с использованием SQL-запросов | Обмен с использованием:   * Регламентированных SQL-запросов (далее – РЗ) – SQL-запросов, выраженных в терминах модели данных, загруженной в ПОДД, и зарегистрированных в Ядре ПОДД под символической мнемоникой, используемой Потребителем данных для выполнения этого запроса; * Произвольных SQL-запросов со структурой запроса и ответа, задаваемой Потребителем данных.   Описание данного вида обмена приведено в разделе 1.4.1. |
| 3 | Обмен с использованием Подписок | Обмен данными в соответствии с предварительно созданной подпиской на изменения сведений и получение:   * уведомлений об изменениях;   Описание данного вида обмена приведено в разделе 1.4.2. |
| 4 | Обмен с использованием запросов к REST-сервису ИС Поставщика | Обмен с использованием запросов к зарегистрированным в ПОДД REST-сервисам ИС Поставщика  Описание данного вида обмена приведено в разделе 1.4.3. |

### Классификация регламентированных запросов

Классификация РЗ приведена в таблице Таблица 2.

Таблица 2 – Классификация РЗ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **По количеству Витрин**, в которых размещены данные, используемые в процессе выполнения запроса | *Простые запросы*  Запросы, которые обращаются к данным, размещённым в одной Витрине Поставщика данных. | | *Распределённые запросы*  Запросы, которые обращаются к связанным данным, размещённым в двух или более Витринах Поставщика данных. | |
| **По типу условий отбора и способу обработки данных** | *Запросы по ключу*  Запросы, предполагающие отбор небольшого количества записей, относящихся к единичному объекту. | *Аналитические запросы*  Запросы, предполагающие обработку значительного количества записей и предполагающие получение агрегатов, а не значений из отдельных атрибутов. | | *Выгрузки*  Запросы, обрабатывающие значительное количество записей, но не выполняющие вычисления, а возвращающие отобранные записи Потребителю данных. |
| **По типу допустимой вариативности** | *Фиксированные запросы*  Запросы, не предусматривающие каких-либо возможностей для Потребителя данных уточнить запрос. | | *Параметризованные запросы*  Запросы, содержащий в своём определении параметры, значения которых Потребитель данных задаёт непосредственно перед выполнением запроса. | |
| **По способу предполагаемого использования** | *Универсальные*  Запросы, атрибутный состав которых не имеет разметки, предполагающей какое-то специальное использование этого запроса. | | *Многомерные*  Запросы, результат которого размечен так, что все возвращаемые атрибуты поделены на измерения и факты, таким образом многомерный запрос подходит для использования в средствах OLAP-анализа как источник данных. | |

Классификация РЗ содержит 24 класса, представляющих собой различные комбинации классифицирующих признаков, приведённых в таблице. Например:

* универсальная простая фиксированная выгрузка;
* многомерный распределённый аналитический фиксированный регламентированный запрос;
* универсальный распределённый параметризованный запрос по ключу.

Значения классификаторов «универсальный», «простой» и «фиксированный» обычно не указываются и принимаются по умолчанию. Например:

* «универсальная изолированная фиксированная выгрузка» – это просто «выгрузка»;
* «многомерный распределённый аналитический фиксированный регламентированный запрос» – это «многомерный распределённый аналитический регламентированный запрос».

## Участники информационного обмена с использованием ПОДД СМЭВ

***Участник взаимодействия*** – это орган или организация, участвующий в информационном обмене через СМЭВ.

Каждый УВ может выступать одновременно как в качестве Поставщика данных, предоставляя свои сервисы и/или данные со своих Витрин и ИС, так и в качестве Потребителя данных, выполняя запросы к другим УВ.

Общая схема взаимодействия участников приведена на рисунке Рисунок 1.

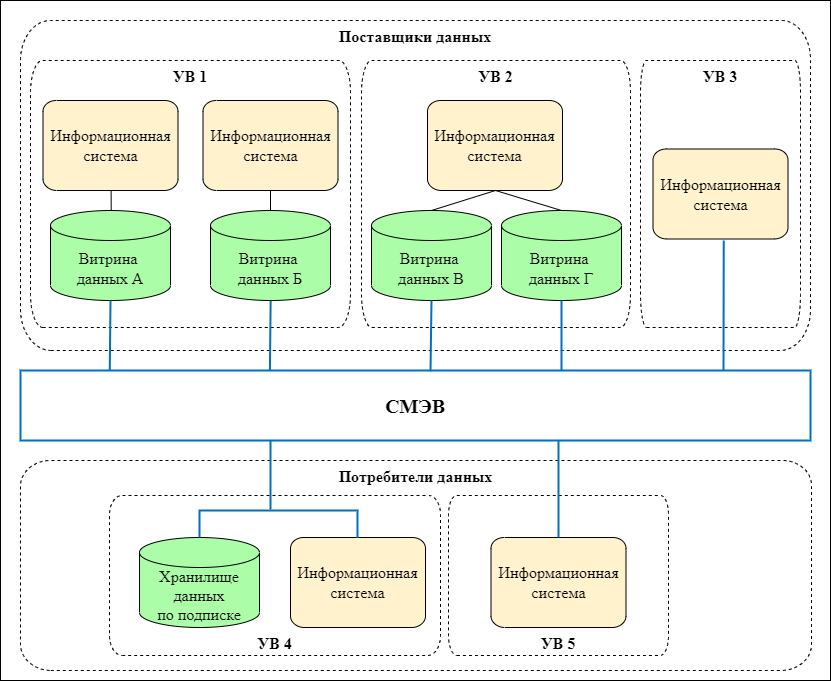


Рисунок 1 – Участники информационного обмена с использованием ПОДД СМЭВ

***Потребители данных*** имеют возможность получать сведения:

1. из Витрины Поставщиков данных следующими способами:
   * используя РЗ (см. п. 3.5 данного документа)
   * используя произвольный SQL-запрос (описание поддерживаемого ПОДД СМЭВ синтаксиса запросов на основе SQL 2003 см. в п. 3.1 данного документа);
   * сформировав подписку на изменения сведений и получая сведения в виде уведомлений об изменениях.
2. из ИС Поставщиков данных (без использования Витрин), выполнив запрос к REST-сервису ИС Поставщика.

***Поставщики данных***:

1. обеспечивают доступ к актуальным данным на Витринах:
   * размещают и актуализируют данные на Витринах;
   * осуществляют подключение Витрин к ПОДД СМЭВ;
2. предоставляют доступ к своим REST-сервисам.

Для обеспечения возможности информационного обмена через ПОДД СМЭВ, УВ должны выполнить требования в части Потребителей данных и Поставщиков данных, приведённые в разделе 1.5 данного документа.

## Информационный обмен участников взаимодействия с использованием ПОДД СМЭВ

Для осуществления возможности информационного обмена с использованием ПОДД СМЭВ должно быть обеспечено подключение ИС УВ к ПОДД СМЭВ с использованием Протокола ПОДД СМЭВ, описание которого приведено в п. 2.1 данного документа.

### Обмен с использованием SQL-запросов

Далее приведено описание информационного обмена с использованием SQL-запросов в ПОДД СМЭВ (общая схема взаимодействия участников обмена и диаграмма последовательности приведены на рисунках Рисунок 2 и Рисунок 3 соответственно).

*В контуре Потребителя данных (УВ 3 или 4 на схемах):*

1. ИС Потребителя данных передаёт SQL-запрос Агенту ПОДД СМЭВ.

Передача запроса может осуществляться с использованием:

* + JDBC-подключения;
  + REST-интерфейса.

Настройки подключения выполняются в рамках настройки ИС для взаимодействия с Агентом ПОДД СМЭВ.

Спецификация взаимодействия приведена в разделе 2.4 данного документа.

1. Агент Потребителя данных проверяет наличие включенного режима блокировки на отправку запросов.
2. В случае отсутствия блокировки Агент ПОДД СМЭВ передает полученный запрос в Ядро ПОДД СМЭВ с использованием Протокола ПОДД СМЭВ.

Спецификация Протокола ПОДД СМЭВ приведена в разделе 2.1.

*Ядро ПОДД СМЭВ (после получения запроса от Агента Потребителя данных):*

1. Выполняет проверку ЭП ОВ, которой подписан запрос, полученный от Агента Потребителя данных.
2. Выполняет проверку наличия у Потребителя данных, отправившего запрос, соответствующих полномочий на выполнение запросов к данным.
3. Выполняет проверку запроса на корректность.

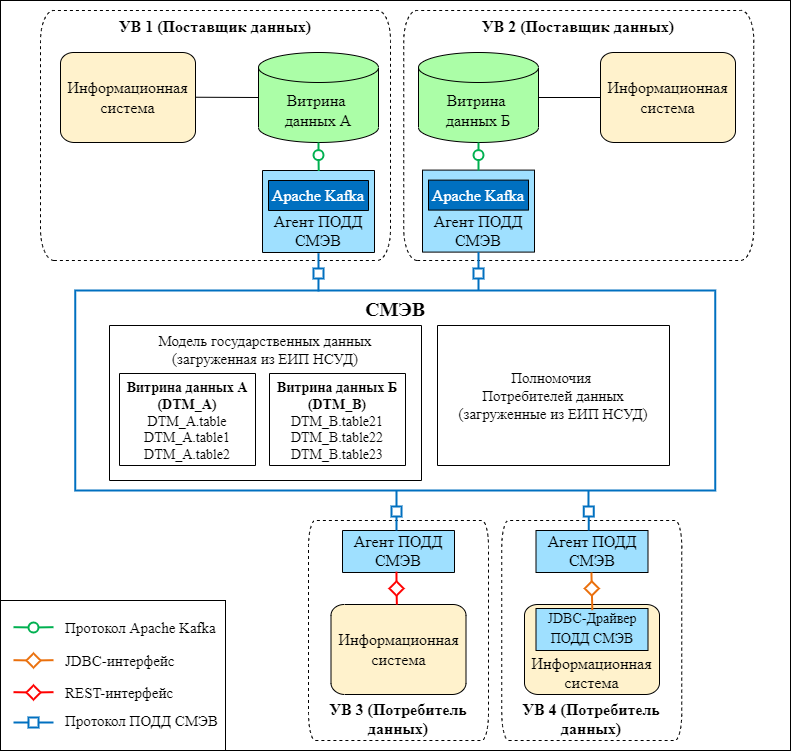


Рисунок 2 – Информационный обмен при выполнении запроса с использованием ПОДД СМЭВ

1. В случае обработки РЗ выполняет преобразование полученной мнемоники РЗ в SQL-выражение в соответствии с загруженным определением РЗ.
2. Выполняет проверку соответствия указанных в запросе атрибутов и объектов Модели государственных данных, загруженной из ФГИС «ЕИП НСУД».
3. Если хотя бы одна из проверок возвращает отрицательный результат, то Ядро ПОДД СМЭВ прекращает выполнение запроса и возвращает Агенту Потребителя данных соответствующую ошибку в качестве результата запроса с использованием Протокола ПОДД СМЭВ.
4. На основании полученного от Потребителя данных запроса Ядро ПОДД СМЭВ формирует один или несколько запросов в адрес Витрин Поставщиков данных (далее – подзапросы).

Рисунок 3 соответствует простому запросу, для которого формируется один подзапрос. Для распределенного запроса Ядро ПОДД СМЭВ формирует два или более подзапроса.

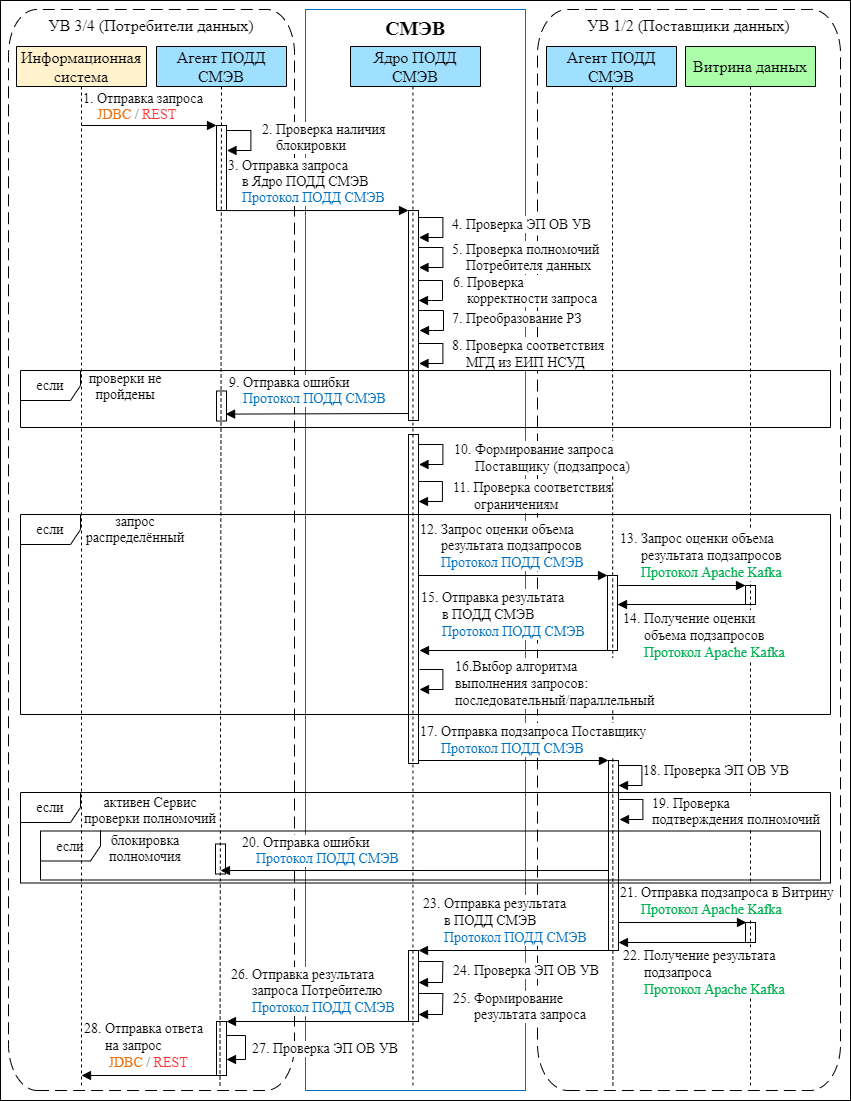


Рисунок 3 – Диаграмма последовательности процесса выполнения SQL-запроса, отправленного от ИС Потребителя данных

1. Выполняет проверку соответствия ограничениям, заданным в Ядре ПОДД СМЭВ:
   * на интенсивность запросов от ИС Потребителя данных за интервал времени;
   * на общий объем данных в запросе и подзапросах за интервал времени.

В случае превышения установленных ограничений Ядро ПОДД СМЭВ прекращает выполнение запроса или подзапроса и возвращает Агенту Потребителя данных сообщение о включении режима блокировки. Агент Потребителя данных при получении такого сообщения активирует режим блокировки и отвечает ошибкой на запросы от Потребителя данных.

*Если запрос распределённый, осуществляется оптимизация запроса:*

1. Ядро ПОДД СМЭВ запрашивает у соответствующих Витрин оценку объема результатов подзапросов.
2. Передача запроса на оценку в Витрину от Агента Поставщика данных осуществляется через Apache Kafka.

Спецификация взаимодействия приведена в разделе 2.3.

1. Возвращение результата от Витрины в Агент Поставщика данных осуществляется таже через Apache Kafka.
2. Ядро ПОДД получает оценку от Агента Поставщика данных с использованием Протокола ПОДД СМЭВ.
3. На основании полученной от Витрин оценки Ядро ПОДД СМЭВ выбирает последовательный или параллельный алгоритм выполнения подзапросов.

*Ядро ПОДД СМЭВ:*

1. Передаёт каждый подзапрос в соответствующий Агент Поставщика данных с использованием Протокола ПОДД СМЭВ.

*В контуре Поставщика данных (УВ 1 и 2 на схемах):*

1. Агент Поставщика данных получает запрос с использованием Протокола ПОДД СМЭВ и проверяет ЭП ОВ, которой подписан запрос.

Дополнительно для простых исходных запросов Потребителя данных в рамках получения подзапроса от Ядра ПОДД в Агент Поставщика передается исходный запрос с ЭП ОВ и сертификатом Потребителя данных, отправившего исходный запрос.

*Если на стороне Агента Поставщика данных развернут и включен Сервис проверки полномочий:*

1. Сервис проверки полномочий выполняет проверку подтверждения полномочия на выполнение регламентированного запроса для Потребителя данных, отправившего запрос.
2. Если для запроса не подтверждено полномочие, то Агент Поставщика прекращает выполнение запроса и возвращает Агенту Потребителя соответствующую ошибку в качестве результата запроса с использованием Протокола ПОДД СМЭВ. При подтвержденном полномочии выполнение запроса продолжается.

*В контуре Поставщика данных (УВ 1 и 2 на схемах):*

1. Агент Поставщика данных помещает запрос в зарезервированный топик брокера сообщений Apache Kafka.
2. Агент Поставщика данных считывает результат выполнения подзапроса из зарезервированного топика брокера сообщений Apache Kafka, формирует ответ и передает его в Ядро ПОДД СМЭВ с использованием Протокола ПОДД СМЭВ.

*Ядро ПОДД СМЭВ:*

1. Ожидает результаты по подзапросам ко всем Витринам Поставщиков данных исходного запроса. Если хотя бы по одному была получена ошибка, Ядро ПОДД СМЭВ прекращает выполнение запроса и возвращает Агенту Потребителя данных соответствующую ошибку в качестве результата запроса с использованием Протокола ПОДД СМЭВ.
2. Осуществляет проверку ЭП ОВ УВ.
3. После получения результатов по всем подзапросам Ядро ПОДД СМЭВ формирует результат для исходного запроса.
4. Отправляет результат Агенту Потребителя данных с использованием Протокола ПОДД СМЭВ.

*В контуре Потребителя данных (УВ 3 или 4 на схемах):*

1. Агент Потребителя данных проверяет ЭП ОВ УВ.
2. Агент Потребителя данных передает ИС с использованием JDBC-подключения или REST-интерфейса результат запроса.

### Обмен с использованием Подписок на уведомления об изменениях

#### Общее описание информационного обмена

ПОДД СМЭВ позволяет организовать процесс автоматического информирования Потребителя данных об изменениях на Витрине Поставщика данных посредством передачи уведомления об изменениях;

Такой обмен осуществляется по предварительно созданной подписке на уведомления об изменениях соответственно.

Загрузка подписки в ПОДД осуществляется через ВС «Подписка на репликацию или уведомлений в изменении данных»[[1]](#footnote-2).

Данный вид обмена в ряде случаев оптимальнее информационного обмена с использованием SQL-запросов и позволяет оптимизировать процесс получения результатов.

Подписка позволяет автоматически загружать изменения данных из Витрины Поставщика в специальное хранилище на стороне Потребителя (Хранилище данных по подписке) и Потребитель посылает запросы напрямую в своё Хранилище, в результате чего сокращается продолжительность сеансов обмена и необходимость "склейки" запросов на стороне ПОДД.

Информационный обмен по подписке состоит из следующих этапов:

1. Регистрация подписки в Витрине Поставщика данных и создание структуры данных в Хранилище Потребителя данных.
2. Актуализация данных посредством передачи пакета дельт от Витрины Поставщика данных в Хранилище Потребителя данных в одном из режимов:
   1. по расписанию (если оно указано в подписке);
   2. по событию об изменении данных (если расписание не указано в подписке).

*В текущей реализации режим по расписанию идёт как дополнение к режиму по событию.*

*Например, в ситуации, когда установлен режим подписки по расписанию и приходит уведомление от Витрины о событии изменений.*

*ПОДД поступит следующим образом:*

* 1. *если уже произведен запрос данных, либо накатывается дельта на поставщике (в рамках расписания) - проигнорирует уведомление;*
  2. *иначе запросит данные и передаст сразу потребителю.*

В пакете дельт для уведомления об изменении отсутствует хэш-сумма.

1. В случае необходимости отключить подписку, осуществляется отмена подписки через ВС "Отмена подписки на репликацию или уведомлений в изменении данных".

Информационный обмен по подписке сопровождается следующими **ограничениями**:

1. Подписка не обновляется. При необходимости изменений формируется новая подписка.
2. В Витрине Потребителе недопустима произвольная запись данных, кроме процесса получения новых дельт от Витрины Поставщика.
3. Механизма ограничения объема пакета отправляемых дельт не предусмотрено.

#### Взаимодействия участников обмена

1. Взаимодействие Ядра ПОДД СМЭВ с Агентами ПОДД СМЭВ осуществляется с использованием Протокола ПОДД СМЭВ (раздел 2.1).
2. Взаимодействие Агента Поставщика данных с Витриной Поставщика данных осуществляется с использованием зарезервированных топиков брокера сообщений Apache Kafka в соответствии со спецификацией:
   1. перечень топиков Apache Kafka приведенный в разделе 2.3.2;
   2. структуры сообщений приведены в разделе 2.3.5.
3. Взаимодействие Агента Потребителя данных с Хранилищем данных по подписке осуществляется с использованием зарезервированных топиков брокера сообщений Apache Kafka в соответствии со спецификацией:
   1. перечень топиков Apache Kafka приведенный в разделе 2.5.1;
   2. структуры сообщений приведены в разделе 2.5.2.

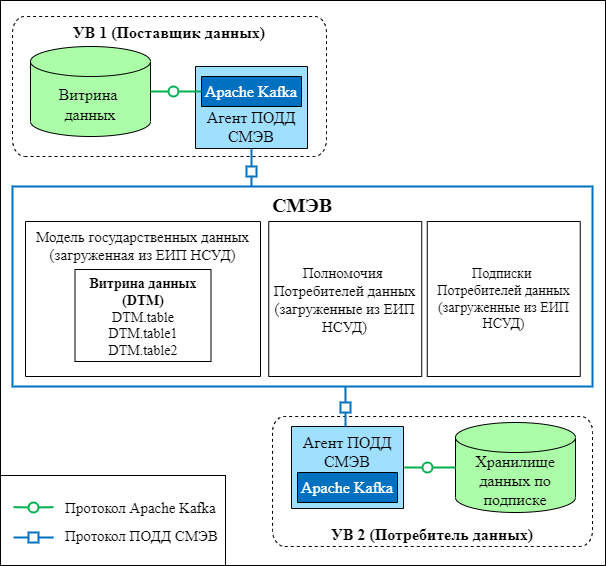


Рисунок 4 – Информационный обмен с использованием подписок на уведомления об изменениях

#### Описание этапов

##### Регистрация подписки

Далее приведён порядок **формирования первоначальной выгрузки данных** (диаграмма последовательности приведена на рисунке Рисунок 5).

*Ядро ПОДД СМЭВ после получения подписки Потребителя данных:*

1. Проверяет наличие новых подписок, загруженных через ВС "Подписка на репликацию или уведомлений в изменении данных".
2. Выполняет проверку наличия у Потребителя данных, указанного в подписке, соответствующих полномочий на доступ к данным, указанным в подписке (достаточно доступа к используемым в запросе атрибутам).

Полномочия должны быть загружены с использованием ВC "Права доступа" до загрузки подписки.

1. В случае наличия прав формирует и отправляет информацию о полученной подписке Агенту Поставщика данных, указанному в подписке.

Иначе формирует ошибку и прекращает обработку подписки. Сведения о подписке, не прошедшей проверку в ПОДД, не хранятся.

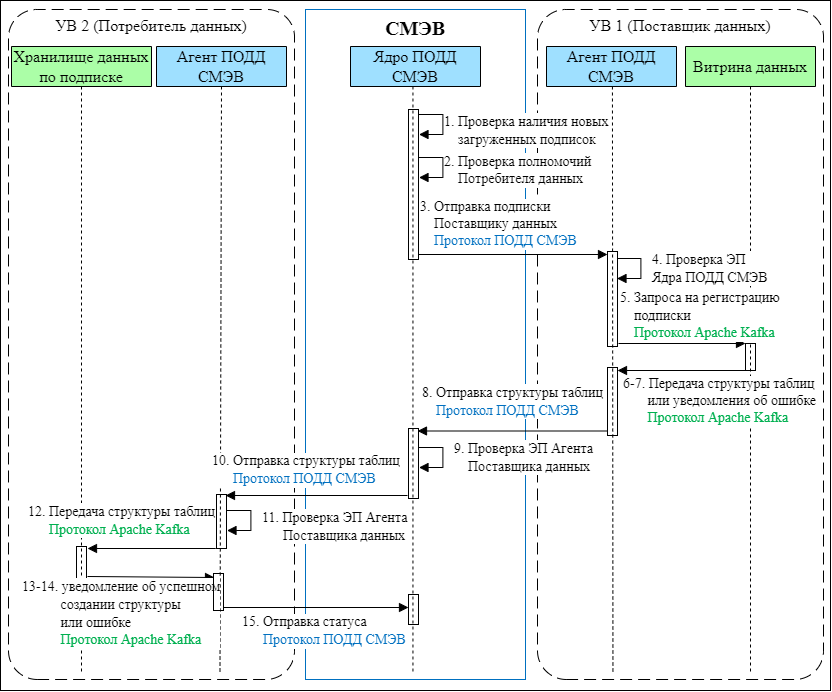


Рисунок 5 – Диаграмма последовательности процесса формирования начальной выгрузки

*Агент Поставщика данных:*

1. Осуществляет проверку ЭП Ядра ПОДД СМЭВ.
2. Передает запрос на регистрацию подписки Витрине в топик *<мнемоника Витрины>.replication.rq.*
3. В случае успешной обработки запроса Витриной, получает структуру таблиц в топик *<**мнемоника Витрины>.replication.rs*.
4. В случае неуспешной обработки запроса получает уведомление об ошибке в топик *<мнемоника Витрины>.replication.err*.
5. Пересылает структуру таблиц в Ядро ПОДД СМЭВ.

*Ядро ПОДД СМЭВ:*

1. Проверяет ЭП Агента Поставщика данных.
2. Пересылает структуру таблиц Агенту Потребителя данных.

*Агент Потребителя данных:*

1. Проверяет ЭП Агента Поставщика данных.
2. Передает в Хранилище данных по подписке структуру таблиц с использованием топика *replication.in.rq*.
3. В случае успешной обработки Хранилищем данных, получает уведомление об успешном создании структуры данных в топик *<мнемоника Витрины>.replication.in.rs*.
4. В случае неуспешной обработки Хранилищем данных, получает уведомление об ошибке в топик *<мнемоника Витрины>.replication.in.err*.
5. Отправляет в Ядро ПОДД СМЭВ статус обработки структуры таблиц Хранилищем данных.

##### Получение пакета дельт

Далее приведён порядок **получения пакета дельт** (диаграмма последовательности приведена на рисунке Рисунок 6).

*Агент Поставщика данных:*

1. **При режиме подписки "по событию об изменении данных":**

Получает уведомление о событии изменения данных на Витрине Поставщика данных в топик *<мнемоника Витрины>.delta.notification*.

**При режиме подписки "по расписанию":**

Данный шаг пропускается, переход к шагу 3 осуществляется по расписанию, указанному в подписке.

1. Пересылает уведомление об изменении данных в Ядро ПОДД СМЭВ.

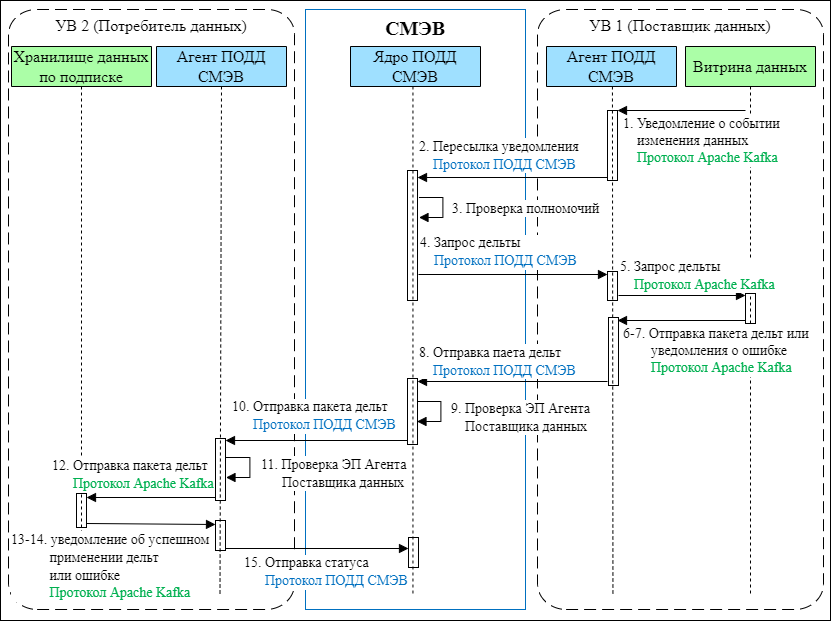


Рисунок 6 – Диаграмма последовательности процесса получения пакета дельт

*Ядро ПОДД СМЭВ:*

3. Проверяет полномочия Потребителей данных, указанных в подписке.

4. В случае наличия прав отправляет запрос пакета дельт Агенту Поставщика данных.

*Если Потребитель еще не прислал ответ о результате применения предыдущего пакета дельт, новый не запрашивается.*

*После того как ответ получен, запрос осуществляется по расписанию в ПОДД.*

В случае отсутствия прав, по расписанию повторно выполняет проверку полномочий.

*Агент Поставщика данных:*

5. Запрашивает пакет дельт у Витрины Поставщика данных через топик *<мнемоника Витрины>.delta.rq*.

*При этом запрос осуществляется:*

* 1. *если предыдущая передача прошла успешно - начиная с synId + 1, где synId - номер последней дельты, принятой в рамках этого пакета дельт;*
  2. *если предыдущая передача прошла неуспешно - начиная с synId, где synId - номер дельты, на которой возникла ошибка.*

*Номер указывается в fromId.*

6. В случае успешной обработки запроса на получение пакета дельт Витриной, получает пакет дельт в топик *<мнемоника Витрины>.delta.rs*.

Ответственность за предоставление правильного пакета дельт (нумерация дельт и наличие хэш-суммы) лежит на Витрине Поставщика данных.

7. В случае неуспешной обработки запроса на получение пакета дельт Витриной, получает уведомление об ошибке в топик *<мнемоника Витрины>.delta.err*.

8. Пересылает пакет дельт в Ядро ПОДД СМЭВ.

*Ядро ПОДД СМЭВ:*

9. Проверяет ЭП Агента Поставщика данных.

10. Пересылает пакет дельт Агенту Потребителя данных.

*Агент Потребителя данных:*

11. Проверяет ЭП Агента Поставщика данных.

12. Передает пакет дельт в Хранилище данных по подписке с использованием топика delta.in.rq.

Хранилище данных по подписке последовательно применяет дельты из пакета.

13. В случае успешной обработки Хранилищем данных, получает уведомление об успешной загрузке пакета дельт в топик *<мнемоника Витрины>.delta.in.rs* с указанием номера последней дельты, принятой в рамках этого пакета дельт.

14. В случае неуспешной обработки Хранилищем данных, получает уведомление об ошибке при загрузке пакета дельт в топик *<мнемоника Витрины>.delta.in.err* с указанием номера дельты, с которой произошла ошибка.

15. Отправляет в Ядро ПОДД СМЭВ статус обработки пакета дельт Хранилищем данных.

##### Отмена подписки

Далее приведён порядок **отмены подписки** (диаграмма последовательности приведена на рисунке [7](#_Отмена_подписки)Рисунок ).

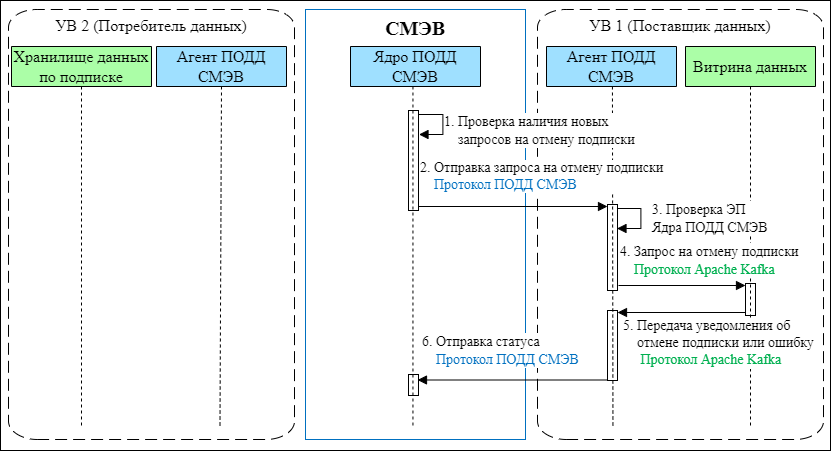


Рисунок 7 – Диаграмма последовательности процесса отмены подписки

*Ядро ПОДД СМЭВ:*

1. Проверяет наличие новых запросов на отмену подписки загруженных через ВС "Отмена подписки на репликацию или уведомлений в изменении данных".
2. Отправляет запрос на отмену подписки в Агент Поставщика данных.

*Агент Поставщика данных:*

1. Осуществляет проверку ЭП Ядра ПОДД СМЭВ.
2. Отправляет запрос Витрине Поставщика в топик *<мнемоника Витрины>.replication.cancel.rq.*
3. Получает ответ от Витрины в топик *<мнемоника Витрины>.replication.cancel.rs*.
4. Отправляет в Ядро ПОДД СМЭВ статус обработки запроса на отмену подписки.

*При этом если от Витрины получен ответ с информацией об ошибке, подписка не отменяется.*

*Данное условие является обязательным, так как если Поставщик не отменит подписку, будут приходить оповещения о наличии изменений по подписке, которые будут вызывать ошибки в ПОДД.*

### Обмен с использованием запросов к REST-сервису ИС Поставщика

#### Общее описание информационного обмена

Обмен с использованием запросов к REST-сервису ИС Поставщика предназначен для сквозного («прозрачного») доступа ИС Потребителей к REST-сервисам ИС Поставщиков. ПОДД СМЭВ используется в качестве канала передачи данных.

Также данный механизм обеспечивает:

1. регистрацию REST-сервисов ИС Поставщиков в ПОДД СМЭВ путём загрузки описания в виде спецификации OpenAPI;
2. разграничение доступа ИС Потребителей к REST-сервисам ИС Поставщиков путём использования принципа выдачи/изъятия прав на взаимодействие ИС Потребителя с REST-сервисом ИС Поставщика;
3. форматно-логический контроль запросов ИС Потребителей и ответов ИС Поставщиков на соответствие спецификации OpenAPI зарегистрированного в ПОДД REST-сервиса.

Основное преимущество использования ПОДД СМЭВ для доступа ИС Потребителя к REST-сервису ИС Поставщика против прямого взаимодействия ИС Потребителя с REST-сервисом ИС Поставщика – сквозной («прозрачный») доступ к REST-сервису за счёт исключения необходимости реализации на стороне ИС Поставщиков и Потребителей:

* защищённого сетевого сегмента (VPN) или ограничения доступа по TLS;
* механизма идентификации, аутентификации и авторизации;
* функций поддержки геораспределённых узлов ИС и балансировки нагрузки (при необходимости);
* механизма ограничения скорости доступа (Rate Limiting).

Максимальный размер запроса при обращении к REST-сервису ИС Поставщика – 5 Мб.

#### Взаимодействие участников обмена

Механизм обмена с использованием запросов к REST-сервису ИС Поставщика реализован в виде комплекса взаимодействующих программных средств и артефактов (Рисунок 8):

1. сервис ИС Потребителя для взаимодействия с REST-сервисом ИС Поставщика;
2. Агент ПОДД, установленный и используемый в инфраструктуре ИС Потребителя;
3. Ядро ПОДД СМЭВ;
4. ЛК УВ, используемый для загрузки и регистрации в ПОДД СМЭВ REST-сервисов ИС Поставщиков;
5. Агент ПОДД, установленный и используемый в инфраструктуре ИС Поставщика;
6. REST-сервис ИС Поставщика, предназначенный к использованию ИС Потребителей;
7. спецификация OpenAPI, описывающая REST-сервис ИС Поставщика.



Рисунок 8 – Информационный обмен с использованием запросов к REST-сервису ИС Поставщика через ПОДД СМЭВ

#### Последовательность осуществления информационного обмена

Подготовка Потребителей и Поставщиков к использованию механизма запросов к REST-сервисам ИС Поставщика описана в разделе 1.5 настоящего документа.

Этапы и основные шаги процесса:

*Разработка и регистрация в ПОДД REST-сервиса ИС Поставщика (Рисунок 9):*

1. Реализация Поставщиком REST-сервиса.
2. Подготовка спецификации OpenAPI в соответствии с реализованным REST-сервисом.  
   Примечание: основные положения по подготовке спецификаций OpenAPI для использования в ПОДД СМЭВ приведены в подразделе 1.6.4 настоящего документа.
3. Регистрация Поставщиком REST-сервиса своей ИС в ПОДД СМЭВ путем загрузки соответствующей спецификации OpenAPI через ЛК УВ.

*Инициализация Агента Потребителя (Рисунок 10):*

1. Агент ПОДД Потребителя, сконфигурированный для работы с механизмом обмена с использованием запросов к REST-сервису Поставщика, уведомляет Ядро ПОДД о своем запуске.
2. Ядро ПОДД СМЭВ передаёт в Агент ПОДД спецификации OpenAPI на REST-сервисы ИС Поставщиков.

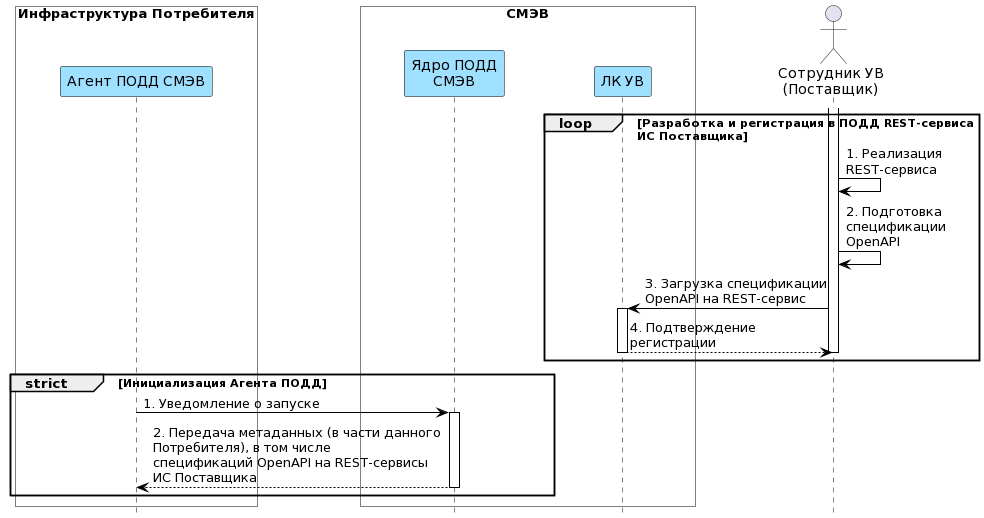


Рисунок 9 – Диаграмма последовательности инициализации информационного обмена с использованием запросов к REST-сервисам ИС Поставщика через ПОДД СМЭВ

*Взаимодействие ИС Потребителя и ИС Поставщика с использованием запросов к REST-сервису ИС Поставщика (Рисунок 10):*

*В контуре Потребителя данных:*

* 1. Запрос к требуемому REST-сервису ИС Поставщика, сформированный в ИС Потребителя данных, передается Агенту ПОДД СМЭВ с использованием REST-интерфейса. Спецификация взаимодействия приведена в разделе 2.4.
  2. Агент Потребителя данных проверяет, что запрос и его параметры соответствуют спецификации OpenAPI REST-сервиса ИС Поставщика и подписывает сообщение ЭП Потребителя.
  3. Агент Потребителя данных передает запрос в Ядро ПОДД СМЭВ с использованием Протокола ПОДД СМЭВ.

*Ядро ПОДД СМЭВ после получения запроса от Агента Потребителя данных:*

* 1. Выполняет проверку ЭП Агента Потребителя, которой подписан запрос.
  2. Выполняет проверку наличия у Потребителя данных, отправившего запрос, соответствующих полномочий на выполнение запросов к REST-сервису.

Если хотя бы одна из проверок возвращает отрицательный результат, то Ядро ПОДД СМЭВ прекращает выполнение запроса и возвращает Агенту Потребителя данных соответствующую ошибку в качестве результата запроса с использованием Протокола ПОДД СМЭВ.

* 1. На основании полученного от Потребителя данных запроса формирует и отправляет запрос Агенту Поставщика данных.

*В контуре Поставщика данных:*

* 1. Агент Поставщика проверяет ЭП Агента Потребителя, которой подписан запрос.
  2. Агент Поставщика извлекает из сообщения запрос и формирует абсолютный URL-адрес запроса к соответствующему методу REST-сервиса ИС Поставщика.
  3. Агент Поставщика отправляет запрос REST-сервису ИС Поставщика данных согласно полученному URL адресу.
  4. REST-сервис ИС Поставщика выполняет полученный запрос (поиск и извлечение требуемых данных, создание или модификация сущностей БД и другое).
  5. REST-сервис ИС Поставщика возвращает в Агент ПОДД ответ на полученный запрос с HTTP кодом статуса выполнения операции и (в зависимости от запрошенного метода) тело ответа, содержащее запрошенные данные и/или дополнительную информацию о результате выполнения операции.
  6. Агент Поставщика формирует ответ (ответ может быть разбит на чанки) и подписывает его ЭП Агента Поставщика.
  7. Агент Поставщика передает ответ в Ядро ПОДД СМЭВ с использованием Протокола ПОДД СМЭВ.

*Ядро ПОДД СМЭВ:*

* 1. Выполняет проверку ЭП Агента Поставщика, которой подписан ответ (чанк ответа).
  2. Перенаправляет результат (чанки результата) Агенту Потребителя данных с использованием Протокола ПОДД СМЭВ.

*В контуре Потребителя данных:*

* 1. Агент Потребителя данных выполняет проверку ЭП Агента Поставщика, которой подписан ответ.
  2. Агент Потребителя данных получает результат запроса, извлекает из сообщения HTTP код и тело (если присутствует) и возвращает их в качестве ответа ИС Потребителя с использованием REST-интерфейса.

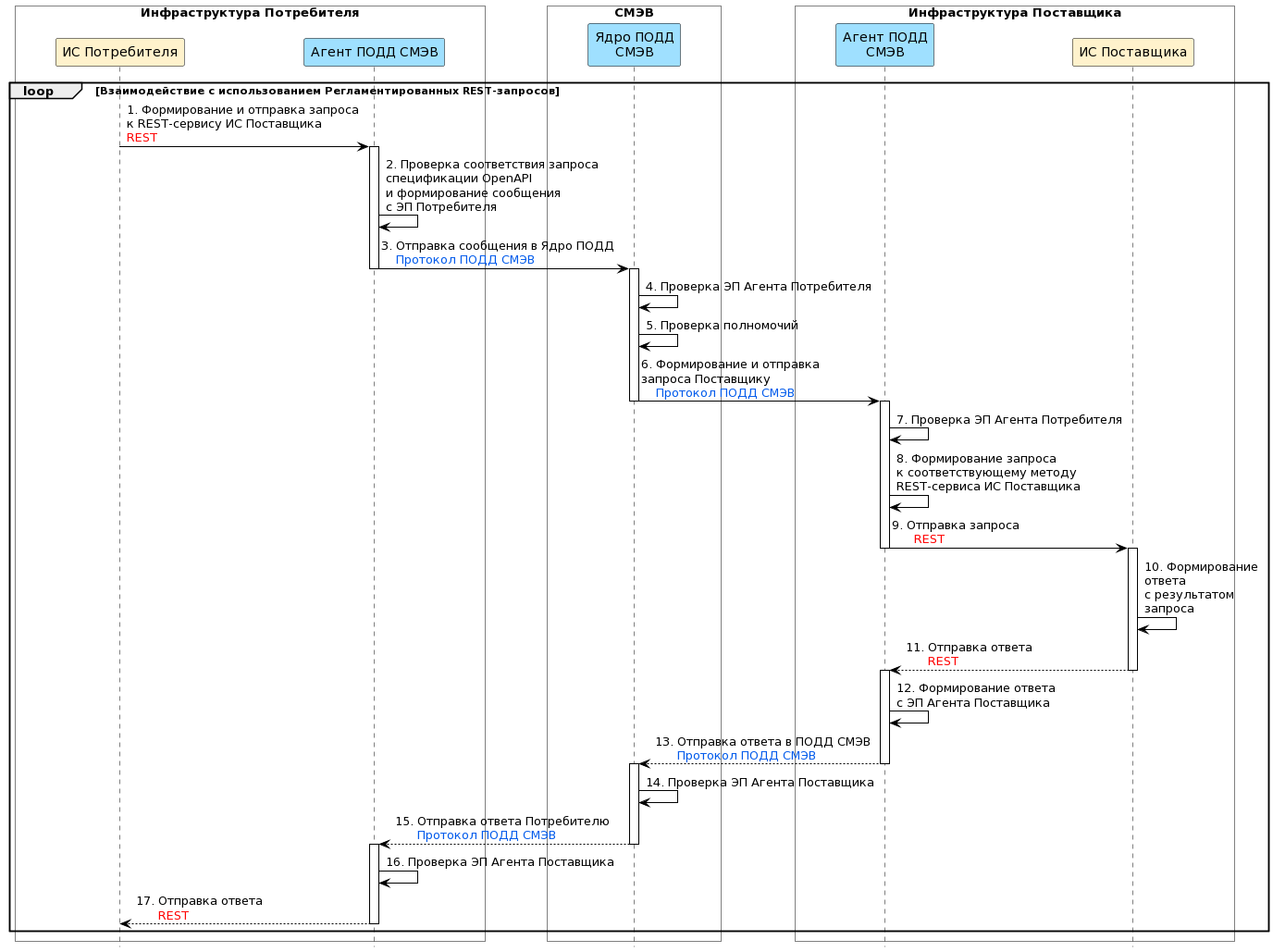


Рисунок 10 – Диаграмма информационного обмена с использованием запросов к REST-сервису ИС Поставщика через ПОДД СМЭВ

## Требования к участникам взаимодействия для использования ПОДД СМЭВ

### Требования к Поставщикам данных

Поставщик данных для участия в информационном обмене с использованием ПОДД СМЭВ должен выполнить требования, перечисленные в таблице Таблица 3.

Таблица 3 – Требования к Поставщикам данных

| **№** | **Требование** | **Вид информационного обмена** | **Комментарий[[2]](#footnote-3)** |
| --- | --- | --- | --- |
|  | Зарегистрировать ИС в СМЭВ | Все | В соответствии с «Регламентом подключения к СМЭВ 4» |
|  | Разместить в своем контуре Агент ПОДД СМЭВ, выполнить настройки для соответствующего вида обменов | Все | В соответствии с «Руководством администратора Агента ПОДД СМЭВ» |
|  | Обеспечить сетевую связанность Агента, размещенного в инфраструктуре участника, и ПОДД СМЭВ | Все |
|  | Разместить в своем контуре Витрины данных | * с использованием SQL-запросов; * с использованием подписок | Исходные коды Витрины размещены в НФАП |
|  | Настроить Витрины для взаимодействия с Агентом | * с использованием SQL-запросов; * с использованием подписок | См. раздел 2.3 данного документа |
|  | Загрузить МГД в части своих данных | Все | В соответствии с «Регламентом подключения к СМЭВ 4» |
|  | Согласовать полномочия по доступу к данным Потребителей данных | Все |
|  | Связать Витрины данных и ИС с использованием ЛК УВ | * с использованием SQL-запросов; * с использованием подписок | В соответствии с инструкцией в «Руководстве пользователя ЛК УВ» |
|  | Зарегистрировать Регламентированные запросы в ПОДД СМЭВ | * с использованием SQL-запросов | В соответствии с «Регламентом подключения к СМЭВ 4» |
|  | Реализовать в своем контуре REST-сервис | С использованием запросов к REST-сервису ИС Поставщика |  |
|  | Зарегистрировать в ПОДД СМЭВ спецификацию OpenAPI, соответствующую реализованному REST-сервису |  |
|  | Добавить критерии доступа для соответствующего информационного обмена. | Все |  |

### Требования к Потребителям данных

Потребитель данных для участия в информационном обмене с использованием ПОДД СМЭВ должен выполнить требования, перечисленные в таблице Таблица 4.

Таблица 4 – Требования к Потребителям данных

| **№** | **Требование** | **Вид информационного обмена** | **Комментарий[[3]](#footnote-4)** |
| --- | --- | --- | --- |
|  | Зарегистрировать ИС в СМЭВ | Все | В соответствии с «Регламентом подключения к СМЭВ 4» |
|  | Разместить в своем контуре Агент ПОДД СМЭВ, выполнить настройки для соответствующего вида обменов | Все | В соответствии с «Руководством администратора Агента ПОДД СМЭВ» |
|  | Обеспечить сетевую связанность Агента, размещенного в инфраструктуре участника, и ПОДД СМЭВ | Все |
|  | Настроить ИС, которая выполняет запросы и осуществляет обработку данных, получаемых от ПОДД СМЭВ, для взаимодействия с Агентом | * с использованием SQL-запросов; * с использованием запросов к REST-сервису ИС Поставщика | См. раздел 2.4 данного документа |
|  | Получить полномочия на доступ к данным, которые требуется обрабатывать в ИС | Все | В соответствии с «Руководством пользователя ЛК УВ» |
|  | Развернуть в своем контуре ПО Хранилища данных по подписке и настроить его взаимодействие с Агентом | с использованием подписок |  |

## Метаданные ПОДД СМЭВ

Метаданные ПОДД СМЭВ содержат информацию о следующих объектах:

1. модели данных Витрин Поставщиков данных;
2. полномочия Потребителей данных;
3. подписки Потребителей данных;
4. определения РЗ;
5. спецификации OpenAPI для REST-сервисов ИС Поставщиков.

Ограничения на формат значений в загружаемых метаданных и сведения о проверках на стороне ПОДД СМЭВ приведены в таблице 5.

Таблица 5 ­– Ограничения на формат значений в метаданных

| **№** | **Характеристика сущности ПОДД СМЭВ** | **Тип** | **Формат** | **Проверки в ПОДД СМЭВ** |
| --- | --- | --- | --- | --- |
| 1 | Идентификатор | uuid | Маска: "([0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12})|(\{[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}\})" | - |
| 2 | Мнемоника | String | Латинские буквы, цифры и символ "\_", без пробелов. Первым символом должна быть буква. Регистр не учитывается | - |
| 3 | Версия (Витрины/РЗ) | SemanticVersion (complexType) | * Major типа int, >= 0 * Minor типа int, >= 0 | Идентификатор Витрины должен быть одинаковый для всех версий Витрины с данной мнемоникой |
| string | <major>.< minor > |  |
| 4 | ОГРН УВ (владельца Витрины/РЗ) | string | *Ожидаемый формат*: только цифры, должен быть одинаковым для всех версий Витрин/РЗ | - |
| 5 | Даты поддержки | dateTime | YYYY-MM-DDThh:mm:ss[.SSS]Z | - |
| integer (int64) | В миллисекундах от эпохи |  |
| 6 | Текстовое описание (Витрины/РЗ) | string | *Ожидаемый формат*: латинские и русские буквы, цифры, печатные символы и символы разметки (пробел, перенос строки) | - |
| 7 | Текст sql запроса (для РЗ) | String | Поддерживаются возможности, указанные в разделе 3.1.1 данного документа | * должны быть указаны Витрины, зарегистрированные в ПОДД; * должна быть явно указана версия используемой Витрины; * не должно использоваться несколько версий одной Витрины в одном SQL-выражении; * для распределенного запроса должно быть указано не менее двух Витрин; * не поддерживается обращение к другим РЗ; * если РЗ содержит табличные параметры, они должны использоваться в SQL-выражении |
| 8 | Тип атрибута | simpleType | * "BOOLEAN" * "STRING" * "BYTE" * "SHORT" * "INTEGER" * "LONG" * "FLOAT" * "DOUBLE" * "BIG\_DECIMAL" * "DATE" * "TIME" * "TIMESTAMP" * "BINARY" | - |
| 9 | Префикс в URL для обмена с использованием запросов к REST-сервисам | string | Латинские буквы, цифры, символы. Из зарезервированных символов допустимы '/' и '\_' | * непустая строка; * начинается с символа '/'; * содержит символы после '/'. Допускается несколько элементов пути, например level1/level2/level3. Двойные слэши '//' недопустимы; * параметры недопустимы, например /{parameter}; * заканчивается символом, отличным от '/' |

### Модель данных Витрины Поставщика данных

Все передаваемые, получаемые и распространяемые посредством ПОДД СМЭВ сведения должны соответствовать моделям данных Витрины Поставщика данных в Ядре ПОДД СМЭВ.

Модели данных загружаются в Ядро ПОДД СМЭВ из ФГИС «ЕИП НСУД».

**Модель данных Витрины Поставщика данных содержит:**

1. уникальный идентификатор Витрины данных;
2. уникальную мнемонику Витрины данных;
3. версии Витрины данных, содержащие структуру сущностей.

**Каждая версия Витрины данных содержит:**

1. мнемонику соответствующей ИС Поставщика данных;
2. ОГРН соответствующего Поставщика данных;
3. номер версии Витрины;
4. срок действия версии Витрины Поставщика данных (дата начала и дата окончания действия версии Витрины);
5. перечень таблиц версии Витрины со следующей информацией:
   * мнемоника и описание таблицы;
   * перечень атрибутов таблицы с указанием типа данных;
   * мнемоника атрибута, являющегося первичным ключом таблицы.

Подробное описание приведено в документации на ВС «Метаданные витрины данных».[[4]](#footnote-5)

**Правила/ограничения использования:**

1. Модель данных Витрины Поставщика данных загружена в Ядро ПОДД СМЭВ, если загружена как минимум одна (первая) версия Витрины.
2. ИС и ОГРН Поставщика данных, а также мнемоника Витрины данных задаются для каждой версии отдельно, поэтому контроль их соответствия у версий ложится на создателей этих версий.
3. Актуальной версией Витрины считается максимальная по номеру среди действующих.
4. Для осуществления доступа к определенной версии Витрины данных необходимо в запросе после мнемоники витрины явно указать номер версии. Пример:

|  |
| --- |
| Запрос:  **select** \* **from** datamart.**3**.table1 **limit 1**  в котором:   * datamart» – мнемоника Витрины данных; * «3» – номер версии; * «table1» – мнемоника таблицы.   Результат: Первая строка из таблицы с мнемоникой table1 Витрины данных datamart версии 3 |

Иначе запросы без указания версии Витрины преобразуются к актуальной версии Витрины. Пример:

|  |
| --- |
| Для Витрины datamart загружены три версии 1, 2 и 3, и версия 3 является актуальной.  Запрос:  **select** \* **from** datamart.table1 **limit 1**  Преобразуется к виду:  **select** \* **from** datamart.**3**.table1 **limit 1** |

1. При формировании в Ядре ПОДД подзапроса к Витрине данных мнемоника Витрины удаляется из запроса, а пара «номер версии витрины – мнемоника сущности» преобразуется к виду v<номер версии>\_<мнемоника сущности>. Пример:

|  |
| --- |
| Для запроса:  **select** \* **from** datamart.**3**.table1 **limit 1**  Подзапрос к Витрине данных будет иметь следующий вид:  **select** \* **from** v3\_table1 **limit 1** |

### Подписки Потребителей данных ПОДД СМЭВ

Подписка предоставляет возможность Потребителю данных оперативно получать изменяющиеся в Витринах Поставщиков данные.

Подписка определяется следующими параметрами:

* уникальный идентификатор подписки;
* источник данных по подписке – мнемоника Витрины Поставщика данных;
* адресат данных по подписке – мнемоника Витрины Потребителя данных;
* набор SQL-выражений, каждое из которых описывает подмножество данных Витрины;
* расписание синхронизации (может отсутствовать).

Функционирование Агента и Ядра ПОДД СМЭВ при формировании первоначальной выгрузки и передачи изменений подписки от Витрины Поставщика в Хранилище данных по подписке приведено в разделах 1.4.2 и 2.5 данного документа.

### Определения Регламентированных запросов

Определения РЗ загружаются в Ядро ПОДД СМЭВ из ФГИС «ЕИП НСУД».

**Состав атрибутов определения РЗ:**

1. уникальный идентификатор РЗ;
2. мнемоника РЗ;
3. версии РЗ.

**Каждая версия РЗ содержит:**

1. описание РЗ (краткое и полное);
2. ОГРН владельца РЗ;
3. номер версии в формате major.minor;
4. срок действия РЗ (дата начала и дата окончания действия РЗ, в рамках которых возможно выполнение соответствующего запроса)
5. SQL-выражение в соответствии с п. 3.1 данного документа.
6. список описаний атрибутов, включающий:
   * мнемоника атрибута;
   * тип атрибута;
   * флаг принадлежности к персональным данным;
   * описание атрибута;
7. список параметров. Порядок параметров в списке соответствует порядку их представления в SQL-выражении. Описание параметра включает в себя:
   * имя параметра;
   * тип параметра;
   * описание параметра;
   * значение по умолчанию.

Подробное описание приведено в документации на ВС «Загрузка регламентированного запроса»[[5]](#footnote-6).

**Правила/ограничения использования:**

1. РЗ не подлежат модификации. При необходимости изменить запрос создается запрос с новой версией.
2. ОГРН Владельца и мнемоника РЗ задаются для каждой версии отдельно, поэтому контроль их соответствия у версий ложится на создателей этих версий.
3. Актуальной версией РЗ считается максимальная по номеру среди действующих.
4. Порядок параметров в списке соответствует порядку их представления в SQL-выражении.

Загрузка РЗ в ПОДД СМЭВ осуществляется с использованием ЛК УВ в соответствии с инструкцией в «Руководстве пользователя ЛК УВ»[[6]](#footnote-7);

Пример определения РЗ:

|  |
| --- |
| {  "content": {  "poddArea": "TEST",  "id": "3544a6a6-d3cb-4602-9099-f891d508f29a",  "queryMnemonic": "datamart\_mnemonic.query\_mnemonic",  "tenantId": "6661228d-af42-4f30-89e1-6b9b8f6eb864",  "tenantOgrn": "1234567890123",  "version": "1.0",  "supportedFromUTC": 1600862481,  "supportedUpToUTC": 1632398481,  "brief": "Краткое описание",  "description": "Полное описание",  "sql": "select colA, colB from datamart\_mnemonic.5.0.tableA",  "columns": [  {  "mnemonic": "colA",  "description": "Колонка A",  "type": "string"  },  {  "mnemonic": "colB",  "description": "Колонка B",  "type": "string"  }  ]  }  }  } |

### Спецификации OpenAPI для REST-сервиса ИС Поставщика

При разработке спецификаций OpenAPI, используемых для описания и регистрации REST-сервисов Поставщиков данных в ПОДД СМЭВ, следует руководствоваться форматом проектирования спецификаций OpenAPI версии 3.0 (в том числе минорных версий 3.0.1, 3.0.2 и т.д.).

Допустимые к загрузке в ПОДД СМЭВ форматы спецификаций OpenAPI, описывающих REST-сервис ИС Поставщиков: YAML или JSON. Максимальный размер загружаемой спецификации OpenAPI 1 Мб.

В спецификациях OpenAPI, описывающих REST-сервисы ИС Поставщиков для ПОДД СМЭВ, могут быть использованы все типы HTTP-методов (POST, GET, PUT и DELETE), выполняющих CRUD-операции (Create/Read/Update/Delete).

Примеры спецификаций OpenAPI, используемых в работе с REST-сервисами в ПОДД СМЭВ:

1) Спецификация OpenAPI, описывающая REST-сервис с функцией возврата запрошенных данных от ИС Поставщика:

|  |
| --- |
| {  "openapi": "3.0.1",  "info": {  "title": "Letters content | Содержимое писем",  "version": "1.0"  },  "servers": [  {  "url": "/"  }  ],  "tags": [  {  "name": "Letters content | Содержимое писем"  }  ],  "paths": {  "/api/v1.0/letters/{mailId}/{date}/pdf": {  "get": {  "tags": [  "Letters content | Содержимое писем"  ],  "summary": "Получение содержимого письма в формате PDF",  "operationId": "getLetterPdf",  "parameters": [  {  "name": "Authorization",  "in": "header",  "description": "Authorization header",  "required": true,  "schema": {  "type": "string"  }  },  {  "name": "mailId",  "in": "path",  "description": "ШПИ",  "required": true,  "schema": {  "type": "string"  }  },  {  "name": "date",  "in": "path",  "description": "Дата",  "required": true,  "schema": {  "type": "string"  }  },  {  "name": "accessCode",  "in": "query",  "description": "Код доступа",  "required": true,  "schema": {  "type": "string"  }  }  ],  "responses": {  "200": {  "description": "ОК",  "content": {}  },  "400": {  "description": "Некорректный запрос",  "content": {}  },  "401": {  "description": "Некорректный идентификатор клиента и/или токен",  "content": {}  },  "403": {  "description": "Доступ запрещен",  "content": {}  },  "404": {  "description": "Запрашиваемый ресурс не найден",  "content": {}  },  "500": {  "description": "Внутренняя ошибка сервиса",  "content": {}  }  }  }  },  "/api/v1.0/letters/{mailId}/{date}/content": {  "get": {  "tags": [  "Letters content | Содержимое писем"  ],  "summary": "Получение содержимого письма в формате ZIP",  "operationId": "getLetterContent",  "parameters": [  {  "name": "Authorization",  "in": "header",  "description": "Authorization header",  "required": true,  "schema": {  "type": "string"  }  },  {  "name": "mailId",  "in": "path",  "description": "ШПИ",  "required": true,  "schema": {  "type": "string"  }  },  {  "name": "date",  "in": "path",  "description": "Дата",  "required": true,  "schema": {  "type": "string"  }  },  {  "name": "accessCode",  "in": "query",  "description": "Код доступа",  "required": true,  "schema": {  "type": "string"  }  }  ],  "responses": {  "200": {  "description": "ОК",  "content": {}  },  "400": {  "description": "Некорректный запрос",  "content": {}  },  "401": {  "description": "Некорректный идентификатор клиента и/или токен",  "content": {}  },  "403": {  "description": "Доступ запрещен",  "content": {}  },  "404": {  "description": "Запрашиваемый ресурс не найден",  "content": {}  },  "500": {  "description": "Внутренняя ошибка сервиса",  "content": {}  }  }  }  }  },  "components": {}  } |

2) Спецификация OpenAPI, описывающая REST-сервис с функцией модификации данных ИС Поставщика:

|  |
| --- |
| {  "openapi": "3.0.2",  "info": {  "description": "Спецификация OpenAPI 3.0 для сервиса записи на приём к врачу",  "version": "1.0",  "title": "Запись на приём к врачу – OpenAPI 3.0"  },  "paths": {  "/booking/book": {  "post": {  "summary": "Бронирование слота",  "description": "Бронирование слота",  "operationId": "book",  "requestBody": {  "description": "Запрос",  "content": {  "application/json": {  "schema": {  "$ref": "#/components/schemas/BookRequest"  }  }  },  "required": true  },  "responses": {  "200": {  "description": "Успешная операция",  "content": {  "application/json": {  "schema": {  "$ref": "#/components/schemas/BookResponse"  }  }  }  },  "400": {  "description": "Неверные параметры"  },  "500": {  "description": "Внутренняя ошибка"  }  }  }  },  "/booking/cancel": {  "post": {  "summary": "Отмена брони",  "description": "Отмена брони",  "operationId": "cancel",  "requestBody": {  "description": "Запрос",  "content": {  "application/json": {  "schema": {  "$ref": "#/components/schemas/CancelRequest"  }  }  },  "required": true  },  "responses": {  "200": {  "description": "Успешная операция",  "content": {  "application/json": {  "schema": {  "$ref": "#/components/schemas/BookResponse"  }  }  }  },  "400": {  "description": "Неверные параметры"  },  "500": {  "description": "Внутренняя ошибка"  }  }  }  }  },  "components": {  "schemas": {  "BookRequest": {  "type": "object",  "properties": {  "bookId": {  "type": "string"  },  "slotId": {  "type": "string"  },  "patient\_id": {  "type": "string"  },  "booking\_type": {  "type": "string"  },  "caseNumber": {  "type": "string"  },  "preliminaryReservation": {  "type": "boolean"  },  "referral\_id": {  "type": "string"  },  "cards\_id": {  "type": "string"  },  "email": {  "type": "string"  },  "mobilePhone": {  "type": "string"  }  },  "required": [  "bookId",  "slotId",  "patient\_id"  ]  },  "BookResponse": {  "oneOf": [  {  "$ref": "#/components/schemas/BookResponseSuccess"  },  {  "$ref": "#/components/schemas/BookResponseError"  }  ]  },  "BookResponseSuccess": {  "type": "object",  "properties": {  "bookExtId": {  "type": "string"  },  "slotId": {  "type": "string"  },  "visitTime": {  "type": "string"  },  "duration": {  "type": "string"  },  "serviceId": {  "type": "string"  },  "organizationId": {  "type": "string"  },  "areaId": {  "type": "string"  },  "queueNumber": {  "type": "string"  },  "pincode": {  "type": "string"  },  "window": {  "type": "string"  },  "status": {  "$ref": "#/components/schemas/Status"  }  },  "required": [  "bookExtId",  "slotId",  "visitTime",  "status"  ]  },  "BookResponseError": {  "type": "object",  "properties": {  "status": {  "$ref": "#/components/schemas/Status"  }  },  "required": [  "status"  ]  },  "CancelRequest": {  "type": "object",  "properties": {  "bookExtId": {  "type": "string"  },  "patient\_id": {  "type": "string"  }  },  "required": [  "bookExtId",  "patient\_id"  ]  },  "Status": {  "type": "object",  "properties": {  "statusCode": {  "type": "integer"  },  "statusMessage": {  "type": "string"  }  },  "required": [  "statusCode"  ]  }  }  }  } |

### Полномочия Потребителя данных ПОДД СМЭВ на доступ к данным

Потребителю данных могут быть предоставлены следующие типы полномочий:

* полномочия на выполнение произвольных SQL-запросов к различным уровням модели данных (с разными типами доступа);
* полномочия на выполнение Регламентированных SQL-запросов;
* полномочия на выполнение запросов к REST-сервису ИС Поставщика.

#### Полномочия на выполнение произвольных SQL-запросов

Для предоставления Потребителю данных возможности выполнения произвольных SQL-запросов к данным, размещенным на Витрине Поставщика данных, необходимо предоставить полномочия на доступ к данным Витрины Поставщика данных.

Полномочия могут быть предоставлены на доступ к следующим уровням модели данных:

* к Витрине Поставщика данных в целом;
* к таблице Витрины Поставщика данных;
* к отдельным атрибутам таблицы.

При этом доступ распространяется на все версии Витрины данных.

Типы доступа к элементам модели данных:

* «Полный» – данные предоставляются в том виде, в котором они описаны в модели данных Витрины;
* «Только статистика» – доступ предоставляется к числовым количественным атрибутам. Потребитель данных с таким уровнем доступа к атрибуту не может получить значения из отдельных строк Витрины данных, а может получать только значения статистических функций (среднее, минимум, максимум и т. д.), при этом количество строк, по которым рассчитывается функция (как для запроса в целом, так и групп при использовании *group by* в запросе), должно быть не менее установленного в настройках уровня доступа минимально возможного количества строк (с целью исключения возможности путём анализа статистики получить значения для отдельных строк);
* «Проверка существования» – для атрибутов с таким типом доступа возможна только проверка наличия значений, известных Потребителю данных, в записях Витрины. Запрещена передача Потребителю данных значений таких атрибутов в ответе на запрос (запрещено обращение к атрибутам вне оператора EXISTS);
* «Проверка соответствия» – для атрибутов с таким уровнем доступа возможна только проверка наличия значений, известных Потребителю данных, в конкретной записи, идентифицированной явно заданным Потребителем данных значением уникального ключа. Запрещена передача Потребителю данных значений таких атрибутов в ответе на запрос (запрещено обращение к атрибутам вне оператора EXISTS).

#### Полномочия на выполнение Регламентированного SQL-запроса

Для предоставления Потребителю данных возможности выполнения РЗ к данным Витрины Поставщика данных необходимо предоставить полномочия на выполнение РЗ. Для выполнения РЗ достаточно наличие полномочий на его выполнение, при этом наличие полномочий по уровням модели данных не проверяется.

Полномочия могут быть предоставлены только на РЗ целиком. Права на атрибуты РЗ не выдаются.

При наличии полномочий на выполнение РЗ Потребитель данных автоматически получает доступ ко всем версиям этого запроса.

При появлении новой версии РЗ в нем не меняются входные или выходные данные. Изменение входных и/или выходных данных производится через создание нового РЗ, а не через выпуск новой его версии.

Выдача полномочий в ПОДД СМЭВ осуществляется с использованием ЛК УВ в соответствии с инструкцией в «Руководстве пользователя ЛК УВ»[[7]](#footnote-8).

Отзыв полномочий на выполнение РЗ в ПОДД СМЭВ осуществляется посредством выполнения REST запроса, где должны быть указаны:

* poddArea ­– среда ПОДД;
* permissionId – идентификатор права доступа.

Пример запроса на отзыв полномочий к РЗ:

|  |
| --- |
| POST /api/v1/lk-uv/deletePermission  Content-Type: application/json  {  "content": {  "poddArea": "TEST",  "permissionId": "3544a6a6-d3cb-4602-9099-f891d508f29a",  }  } |

Использование REST запроса для управления полномочиями на выполнение РЗ является временным решением, до полного перехода на управление с использованием ЛК УВ.

Поставщику данных доступна донастройка полномочий на выполнение РЗ с использованием Сервиса проверки полномочий, развернутого на Агенте ПОДД СМЭВ. Данный сервис обеспечивает возможность подтвердить или временно заблокировать полномочия на выполнение версии РЗ, не изменяя загруженные в ПОДД СМЭВ полномочия. Описание порядка настройки и установки Сервиса проверки полномочий приведена в Руководстве администратора Агента ПОДД СМЭВ[[8]](#footnote-9).

#### Полномочия на выполнение запросов к REST-сервису ИС Поставщика

Полномочия на выполнение запросов к REST-сервису ИС Поставщика предоставляются на все запросы, указанные в загруженной в ПОДД СМЭВ спецификации OpenAPI для соответствующего REST-сервиса ИС Поставщика. Права на использование отдельных методов и запросов REST-сервиса не выдаются.

Управление полномочиями осуществляется через запрос в СЦ в соответствии с инструкцией «Регламент подключения к СМЭВ 4[[9]](#footnote-10)».

## Типы данных ПОДД СМЭВ

Поддерживаемые ПОДД СМЭВ типы данных и соответствующие им физические и логические типы данных avro приведены в таблице 6. Типы данных приведены к используемым в системе Prostore (каноническое avro).

Таблица 6 – Соответствие типов данных ПОДД СМЭВ и типов данных avro

| **№** | **Описание типа** | **Размер-ность** | **Имя типа ПОДД** | **Возможные значения** | **Физический тип avro** | **Логический тип avro** | **Особенности соответствия  типов** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Строка |  | STRING | не ограничено | string |  |  |
| 2 | Целое число | 8 байт | LONG | от −9223372036854775808 до 9223372036854775807 | long |  |  |
| 3 | Целое число | 1 байт | BYTE | от −128 до 127 | int |  |  |
| 4 | Целое число | 2 байта | SHORT | от −32768 до 32767 | int |  |  |
| 5 | Целое число | 4 байта | INTEGER | от -2147483648 до 2147483647 | int |  |  |
| 6 | Дата | 4 байта | DATE | от -2147483648 до 2147483647 | int | date | unix epoch в днях, от -5 883 516 года до 5 883 515 года |
| 7 | Время | 8 байт | TIME | от −9223372036854775808 до 9223372036854775807 | long | time-micros |  |
| 8 | Числовые данные с плавающей запятой | 4 байта | FLOAT | от –(2-2-23)·2127 до  (2-2-23)·2127 | float |  |  |
| 9 | Числовые данные с плавающей запятой | 8 байт | DOUBLE | от -1.7\*10308 до 1.7\*10308 | double |  |  |
| 10 | Большие числовые данные с плавающей запятой |  | BIG\_DECIMAL | не ограничено | string | BigDecimalLogicalType |  |
| 11 | Временная метка | 8 байт | TIMESTAMP | от −9223372036854775808 до 9223372036854775807 | long | timestamp-micros  или  LocalDateTimeLogicalType | unix epoch в микросекундах, от  – 292277 года до 292277 года |
| 12 | Двоичные данные |  | BINARY | не ограничено | bytes  или  record |  |  |
| 13 | Булевый тип | 1 байт | BOOLEAN | true, false | boolean |  |  |

# Описание подключения к ПОДД СМЭВ

## Протокол ПОДД СМЭВ

Взаимодействие Агентов ПОДД СМЭВ и Ядра ПОДД СМЭВ реализовано c использованием Протокола ПОДД СМЭВ – бинарного протокола Apache Pulsar с предварительной аутентификацией.

Бинарный протокол Apache Pulsar – протокол, реализующий команды, которыми обмениваются между собой клиент (Агент ПОДД СМЭВ) и брокер Apache Pulsar в составе ПОДД СМЭВ. Команды имеют формат, базирующийся на Protocol Buffers (protobuf). Набор данных, пересылаемый ПОДД СМЭВ по этому протоколу, кодируется в формат Apache Avro.

В рамках предварительной аутентификации клиенты (Агенты ПОДД СМЭВ), устанавливающие подключение по бинарному протоколу Apache Pulsar, получают аутентификационный токен с использованием OpenID Connect (OIDC) – протокола, базирующегося на Oauth 2.0.

## Подключение участников взаимодействия с использованием ПОДД СМЭВ

Основным способом направления обращений является использование Личного кабинета СЦ <https://sc.minsvyaz.ru>.

Электронная почта [sd@sc.minsvyaz.ru](mailto:sd@sc.minsvyaz.ru) является резервным способом направления обращений, который используется в случае недоступности Личного кабинета СЦ.

## Протокол взаимодействия Агента ПОДД СМЭВ и Витрины Поставщика данных

Агент ПОДД СМЭВ Поставщика данных может взаимодействовать с несколькими Витринами данных. Протокол коммуникации Агента и Витрин реализован в виде обмена сообщениями с использованием зарезервированных топиков брокера сообщений Apache Kafka. Брокер сообщений является частью Агента.

### Перечень заголовков сообщений в Apache Kafka

При обработке всех видов сообщений Витрины должны прокидывать заголовки Kafka сообщения из запроса в соответствующий ответ, как в успешный, так и в ошибочный.

Перечень заголовков, которые необходимо прокидывать приведены в таблице Таблица 7

Таблица 7 – Перечень заголовков Apache Kafka

| **Заголовок** | **Описание** |
| --- | --- |
| REQUEST\_ID | Идентификатор запроса |
| AGENT\_CONSUMER\_ID | Идентификатор Агента ПОДД (соответствует мнемонике ИС) |
| QUERY\_MNEMONIC | <Полная мнемоника РЗ>.<версия РЗ> |
| TIMESTAMPS | Временная метка с описанием событий, представляет собой склейку сериализованных в JSON структур типа TimestampData (таблица Таблица 8), разделённых точкой с запятой  Пример:  {"name":"REQUEST\_RECEIVED\_BY\_AGENT\_CONSUMER", "timestamp":1654590839986,"source":"test","result":"UNDEFINED"}; {"name":"REQUEST\_RECEIVED\_BY\_AGENT\_CONSUMER", "timestamp":1654590842487,"source":"test","result":"SUCCESS"} |
| CORE\_ID | Идентификатор Ядра |

Таблица 8 – Структура заголовка TIMESTAMPS

|  |  |
| --- | --- |
| **Атрибут** | **Описание** |
| name | Название временной метки:   * "REQUEST\_RECEIVED\_BY\_AGENT\_CONSUMER" - Запрос поступил в агент-потребитель * "REQUEST\_SENT\_TO\_DATAMART" - Запрос передан из агента-поставщика в витрину * "RESULT\_RECEIVED\_BY\_AGENT\_PRODUCER" - Получен результат из витрины * "RESULT\_SENT\_TO\_AGENT\_CONSUMER" - Обработка в ядре завершена (перед отправкой результата потребителю (Executor/Strom)) |
| timestamp | UTC время в миллисекундах |
| source | Сервис, зафиксировавший временную метку |
| result | Результат обработки:   * UNDEFINED * SUCCESS * SUCCESS\_REQUEST * ERROR * SUCCESS\_ESTIMATE * FAILURE |

### Перечень топиков брокера сообщений Apache Kafka

Конфигурация Агента содержит перечень Витрин данных, с которыми он взаимодействует. Каждой Витрине соответствует свой независимый набор топиков. Полное название топиков формируются по следующему шаблону *<мнемоника Витрины>.<название топика>*.[[10]](#footnote-11)

Названия топиков брокера сообщений приведена в таблице Таблица 9 (названия топиков приведены в качестве примера и могут быть изменены в конфигурации Агента ПОДД СМЭВ). В разделе 2.3.5 приведены структуры сообщений для обмена данными с использованием топиков.

Таблица 9 – Названия топиков брокера сообщений Apache Kafka

| № | Топик | Публикатор | Подписчик | Передаваемый объект |
| --- | --- | --- | --- | --- |
| *Топики регистрации и настройки Витрин* | | | | |
|  | datamart.signal | Витрина | Агент | Сообщение с регистрационными данными Витрины |
|  | <мнемоника Витрины>.profile.rq | Агент | Витрина | Запрос профиля Витрины |
|  | <мнемоника Витрины>.profile.rs | Витрина | Агент | Профиль Витрины |
|  | <мнемоника Витрины>.profile.err | Витрина | Агент | Ошибка при выполнении запроса профиля Витрины |
|  | <мнемоника Витрины>.signal | Агент | Витрина | Сообщение с итогами выполнения регистрации Витрины |
| *Топики для обеспечения информационного обмена с использованием SQL-запросов* | | | | |
|  | <мнемоника Витрины>.query.rq | Агент | Витрина | Подзапрос |
|  | <мнемоника Витрины>.procedure.query.rq | Агент | Витрина | Регламентированный запрос на исполнение |
|  | <мнемоника Витрины>.query.tp | Агент | Витрина | Подзапрос с использованием табличного параметра |
|  | <мнемоника Витрины>.query.rs | Витрина | Агент | Результата подзапроса |
|  | <мнемоника Витрины>.query.estimation.rs | Витрина | Агент | Оценка (статистика) по исполнению подзапросов |
|  | <мнемоника Витрины>.query.err | Витрина | Агент | Ошибка при выполнении подзапроса |
|  | <мнемоника Витрины>.blob.rq | Агент | Витрина | Запрос двоичных данных по ссылке |
|  | <мнемоника Витрины>.blob.rs | Витрина | Агент | Результат запроса двоичных данных по ссылке |
|  | <мнемоника Витрины>.blob.err | Витрина | Агент | Ошибка при выполнении запроса двоичных данных по ссылке |
|  | <мнемоника Витрины>.cancel.rq | Агент | Витрина | Идентификатора запроса, выполнение которого в Витрине нужно отменить |
|  | <мнемоника Витрины>.cancel.rs | Витрина | Агент | Результат успешной отмены запроса |
|  | <мнемоника Витрины>.cancel.err | Витрина | Агент | Ошибка при выполнении отмены запроса |
| *Топики для обеспечения информационного обмена с использованием подписок* | | | | |
|  | <мнемоника Витрины>.replication.rq | Агент | Витрина | Информация о подписке |
|  | <мнемоника Витрины>.replication.rs | Витрина | Агент | Структура данных для хранения данных по подписке |
|  | <мнемоника Витрины>.replication.err | Витрина | Агент | Ошибка при обработке подписки |
|  | <мнемоника Витрины>.delta.rq | Агент | Витрина | Запрос пакета дельт по подписке |
|  | <мнемоника Витрины>.delta.rs | Витрина | Агент | Дельта по подписке |
|  | <мнемоника Витрины>.delta.err | Витрина | Агент | Ошибка при формировании Витриной дельты по подписке |
|  | <мнемоника Витрины>.replication.cancel.rq | Агент | Витрина | Идентификатора отменяемой подписки |
|  | <мнемоника Витрины>.replication.cancel.rs | Витрина | Агент | Результат (успешный или ошибка) отмены подписки |
|  | <мнемоника Витрины>.delta.notification | Витрина | Агент | Уведомление о наличии дельты по подписке в Витрине Поставщика |
| *Топики для получения статистики по Витринам* | | | | |
|  | <мнемоника Витрины>.statistics.rq | Агент | Витрина | Запрос статистики таблиц |
|  | <мнемоника Витрины>.statistics.rs | Витрина | Агент | Статистика таблиц |
|  | <мнемоника Витрины>.statistics.err | Витрина | Агент | Ошибка формировании статистики таблиц |
| *Топики для получения событий Витрины* | | | | |
|  | <мнемоника Витрины>.scl.signal | Витрина | Агент | События Витрины для дальнейшей передачи в СЦЛ |

### Настройка Агента ПОДД для работы с несколькими Витринами данных

В ПОДД предусмотрено два способа регистрации Витрин данных:

* на основе сообщений от Витрин;
* на основе конфигурационного файла.

Описание процесса регистрации Витрины приведено в таблице Таблица 10, схема регистрации на основе сообщения от Витрины приведена на рисунке Рисунок 11.

Регистрация Витрины осуществляется вне зависимости от наличия метаданных, полученных из ЕИП НСУД (п. 1.6.1). Выполнение запросов к Витринам осуществляется только после получения метаданных.

Таблица 10 – Процесс регистрации Витрины данных

|  | **Шаг** | **Регистрация на основе сообщений от Витрин данных** | **Регистрация на основе конфигурационного файла** |
| --- | --- | --- | --- |
| **Инициация регистрации** | 1 | Витрина данных направляет в Агент сообщение с регистрационными данными в общий топик Kafka ***datamart.signal*** предназначенный для получения запросов на регистрацию от всех Витрин. Структура сообщения приведена в таблице Таблица 11  При отправке запроса на регистрацию Витрина должна обеспечить подписку на топик *<мнемоника Витрины>.signal* | При запуске Агент вычитывает локальный конфигурационный файл с предустановленными параметрами Витрин.  В случае наличия топиков для всех Витрин данных в конфигурационном файле, шаги 2-8 пропускаются |
| **Регистрация** | 2 | Агент создаёт необходимые для работы с новой Витриной данных топики в соответствии с таблицей Таблица 9. | |
| 3 | Агент запрашивает у Витрины профиль через топик *<мнемоника Витрины>.profile.rq*. Структура сообщения приведена в таблице Таблица 13 | Агент загружает профиль Витрины. Способ загрузки определяется значением registrationFlow в конфигурационном файле:   * DEFINED\_PROFILE – загрузка по указанному в конфигурации пути; * DATAMART\_REQUEST – запрос к Витрине через топик *<мнемоника Витрины>.profile.rq* |
| 4 | Агент получает:   * либо профиль Витрины через топик *<мнемоника Витрины>.profile.rs* (структура приведена в таблице Таблица 14), * либо сообщение об ошибке в топик *<мнемоника Витрины>.profile.err* (структура приведена в таблице Таблица 15) | |
| 5 | Агент:   * уведомляет Ядро ПОДД СМЭВ о регистрации профиля новой Витрины, * передаёт в Ядро ПОДД СМЭВ профиль Витрины * ожидает подтверждения регистрации от Ядра ПОДД | |
| 6 | Агент получает подтверждение регистрации профиля Витрины от Ядра ПОДД СМЭВ.  При успешном выполнении всех регистрационных операций, либо если данная Витрина уже зарегистрирована возвращается статус «успех», в противном случае «неуспех» | |
| 7 | Агент направляет Витрине сообщение о статусе регистрации. Структура сообщения приведена в таблице Таблица 12 | |
| 8 | В случае неуспешной регистрации Агент удаляет созданные на шаге 2 топики в Kafka | |
| **Действия после успешной Регистрации** | 9 | Агент выполняет запрос к Ядру ПОДД СМЭВ за актуальной информацией о зарегистрированных Витринах (метаданные Витрины и иные данные, необходимые для корректной работы)[[11]](#footnote-12) и обновляет по полученным данным хранимую локально сводную информацию.  Агент завершает процедуру запуска только после получения актуальных данных | |
| **Действия после неуспешной Регистрации** | 10 | В случае ошибок, таймаутов и сбоев, инициация повторной регистрации обеспечивается Витриной | Агент завершает работу с ошибкой |

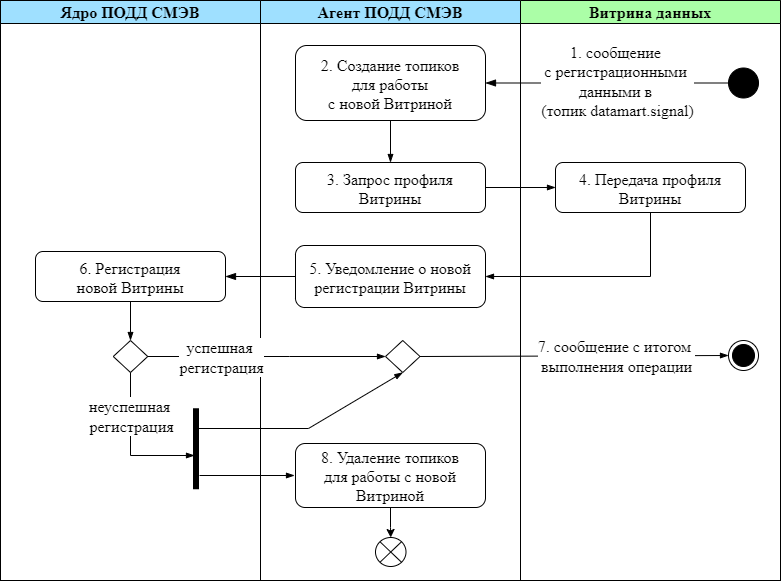


Рисунок 11 – Процесс регистрации Витрины данных на основе сообщения от Витрины данных

### Последовательность исполнения SQL-запроса в контуре Поставщика данных

В контуре Поставщика данных Агент, получив подзапрос от Ядра ПОДД СМЭВ, инициирует выполнение подзапроса Витриной путем отправки сообщения в топик:

* **query.rq** – для произвольных SQL-запросов;
* **procedure.query.rq** – для РЗ;
* **query.tp** – передача чанков табличных параметров к подзапросу.

На рисунке Рисунок 12 приведена последовательность исполнения SQL-запроса в контуре Поставщика данных. Структура сообщений приведена в разделе 2.3.5.

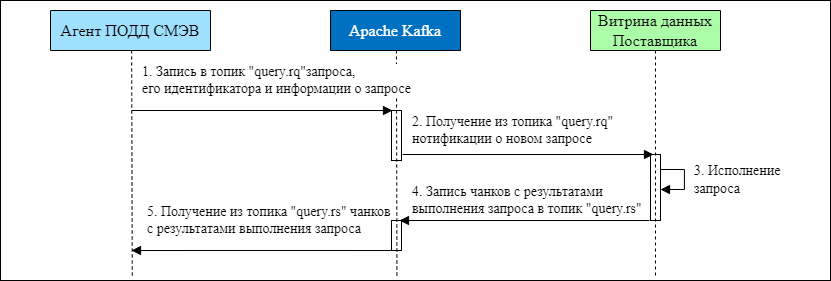


Рисунок 12 – Последовательность исполнения SQL-запроса в контуре ИС Поставщика данных

### Структуры сообщений для взаимодействия с Поставщиком

#### Общие требования

Необязательность атрибута означает возможность указания в качестве значения null. Сам атрибут в сообщении должен присутствовать.

#### Регистрация Витрины данных

##### Регистрация Витрины

###### Запрос на регистрацию Витрины (топик datamart.signal)

Таблица 11 – Структура запроса регистрации Витрины данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "datamartId": string  "timestamp": long  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | Уникальный идентификатор запроса |
| А-2 | datamartId | string | Да | Уникальный идентификатор Витрины. В дальнейшем используется в качестве префикса в топиках обмена данными с Витриной |
| А-3 | timestamp | long | Да | Метка времени формирования запроса Витриной |

###### Ответ на запрос на регистрацию Витрины (топик – <мнемоника Витрины>.signal)

Таблица 12 – Структура ответа на запрос регистрации Витрины данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "datamartId": String  "timestamp": long  "result": String  "errorDescription": String  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | Уникальный идентификатор запроса |
| А-2 | datamartId | string | Да | Уникальный идентификатор Витрины, для которого выполнялась регистрация |
| А-3 | timestamp | long | Да | Метка времени формирования ответа Агентом |
| А-4 | result | string | Да | Результат регистрации. Возможные значения SUCCESS, FAILURE |
| А-5 | errorDescription | string | Да, в случае result=FAILURE | Описание ошибки |

##### Получение профиля Витрины

###### Запрос профиля Витрины (топик - <мнемоника Витрины>.profile.rq)

Таблица 13 – Структура запроса профиля Витрины данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "agentId": string  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | Уникальный идентификатор запроса |
| А-2 | agentId | string | Да | Идентификатор Агента ПОДД |

###### Ответ на запрос профиля Витрины (топик - <мнемоника Витрины>.profile.rs)

Таблица 14 – Структура ответа на запрос профиля Витрины данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "agentId": string  "datamartId": string  "profile":  "sqlDialect": POSTGRESQL  "parameterNotionType": DOLLAR\_NUM  "aggFunctions": [ ]  "otherFunctions": [ ]  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | Уникальный идентификатор запроса |
| А-2 | agentId | string | Да | Идентификатор Агента ПОДД |
| А-3 | datamartId | string | Нет | Уникальный идентификатор Витрины |
| А-4 | profile | DatamartProfile | Да | Профиль Витрины данных |
| А-4.1 | sqlDialect | string | Да | SQL диалект |
| А-4.2 | parameterNotionType | enum | Да | Тип представления параметров в SQL-выражении.  Возможные значения:   * QUESTION (параметр представляется “?”) * DOLLAR\_NUM (параметр представляется “$n”). |
| А-4.3 | aggFunctions | array of string | Да | Перечень функций агрегации |
| А-4.4 | otherFunctions | array of string | Да | Перечень остальных функций |

###### Ответ на запрос профиля Витрины в случае ошибки (топик - <мнемоника Витрины>.profile.err)

Таблица 15 – Структура ответа на запрос профиля Витрины данных в случае ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "errorCode": string  "message": string  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | Уникальный идентификатор запроса |
| А-2 | errorCode | string | Да | Код ошибки |
| А-3 | message | string | Да | Сообщение об ошибке |

#### Информационный обмен с использованием SQL-запросов

##### Получение данных из Витрины

###### Запрос данных из Витрины (<мнемоника Витрины>.query.rq / <мнемоника Витрины>.procedure.query.rq)

Таблица 16 – Структура запроса данных из Витрины

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "subRequestId": UUID  "replyTo": String  "datamartMnemonic": String  "sql": String  "parameters": Array [  {  "type": String  "value": String  }  ]  "tableParams”: Array [  {  "id": UUID  "name": String  "columns": Array [  {  "name": String  "type": String  }  ]  }  ]  "isForEstimation": Boolean  "rowCountThreshold": Long  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | Идентификатор исходного SQL-запроса |
| А-2 | subRequestId | UUID | Да | Идентификатор SQL-подзапроса |
| А-3 | replyTo | String | Да | Служебная информация маршрутизации сообщения.  Ответ, формируемый Витриной, обязан содержать переданное значение без каких-либо искажений. |
| А-4 | datamartMnemonic | String | Да | Мнемоника Витрины, к которой выполняется запрос |
| А-5 | sql | String | Нет | SQL-запрос на исполнение   * При наличии используется топик <мнемоника Витрины>.query.rq. * При отсутствии рассматривается как вызов процедуры с именем, определенным мнемоникой, и опциональными параметрами и используется топик <мнемоника Витрины>.procedure.query.rq |
| А-6 | parameters | Array of object | Да | Параметры к SQL-запросу, использующиеся при выполнении процедур |
| А-6.1 | type | String | Да | Тип параметра |
| А-6.2 | value | String | Нет | Значение параметра |
| А-7 | tableParams | Array of object | Нет | Табличные параметры запроса |
| А-7.1 | id | UUID | Да | Идентификатор табличного параметра |
| А-7.2 | name | String | Да | Имя табличного параметра |
| А-7.3 | columns | Array of object | Да | Информация об атрибутах (колонках) табличного параметра |
| А-7.3.1 | name | String | Да | Имя атрибута |
| А-7.3.2 | type | String | Да | Тип атрибута [ STRING, LONG, INTEGER, BIG\_DECIMAL, DOUBLE, FLOAT, DATE, TIMESTAMP, BOOLEAN, BINARY] |
| А-8 | isForEstimation | Boolean | Да | Признак необходимости выполнить оценку объема и количества строк по данному запросу без отправки результата выполнения запроса |
| А-9 | rowCountThreshold | Long | Да | Количество строк, при не превышении которого Витрина должна выслать результат выполнения сразу же, без отправки оценки объема и количества строк |

###### Передача чанков табличных параметров (<мнемоника Витрины>.query.tp)

Таблица 17 – Структура передачи чанков табличных параметров

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "tableParamId": UUID  "subRequestId": UUID  "chunkNum": Integer  "isLast": Boolean  "streamNum": Integer  "streamTotal": Integer  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | Идентификатор исходного SQL-запроса |
| А-2 | tableParamId | UUID | Да | Идентификатор табличного параметра |
| А-3 | subRequestId | UUID | Да | Идентификатор SQL-подзапроса |
| А-4 | chunkNum | Integer | Да | Номер чанка. В каждом стриме нумерация начинается с 1. |
| А-5 | isLast | Boolean | Да | Признак последнего чанка. У каждого стрима есть последний чанк |
| А-6 | streamNum | Integer | Нет | Номер стрима, нумерация начинается с 1, в итоговой общей последовательности чанков стримы идут последовательно в соответствии с их номерами |
| А-7 | streamTotal | Integer | Нет | Всего стримов |

###### Ответ на запрос - результаты выполнения подзапроса (<мнемоника Витрины>.query.rs)

Таблица 18 – Структура передачи ответа на запрос – результат выполнения запроса

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "subRequestId": UUID  "streamNumber": Integer  "streamTotal": Integer  "chunkNumber": Integer  "isLastChunk": Boolean  "replyTo": String  "isFragmented": Boolean  "uncompressedSize": Integer  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | UUID базового запроса, пришедшего в ПОДД |
| А-2 | subRequestId | UUID | Да | UUID подзапроса, выделенного из базового, SQL-выражение которого передается Витрине |
| А-3 | streamNumber | Integer | Нет | Не заполняется |
| А-4 | streamTotal | Integer | Нет | Не заполняется |
| А-5 | chunkNumber | Integer | Да | Номер чанка |
| А-6 | isLastChunk | Boolean | Да | Признак последнего чанка |
| А-7 | replyTo | String | Да | Служебное поле, формируется из запроса |
| А-8 | isFragmented | Boolean | Да | Признак присутствия в чанке неполных строк (строк, которые были разбиты на несколько чанков) |
| А-9 | uncompressedSize | Integer | Да | Объем чанка до сжатия, в байтах |

В теле сообщения передается чанк сериализованного результата выполнения запроса.  
Сериализация производится в avro с динамической схемой.

###### Ответ на запрос - оценка результатов выполнения подзапроса (<мнемоника Витрины>.query.estimation.rs)

Таблица 19 – Структура передачи ответа на запрос-оценка результатов выполнения подзапроса

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "subRequestId": UUID  "estimatedSize": Long  "estimatedRowCount": Long  "estimatedTime": Long  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | UUID базового запроса, пришедшего в ПОДД |
| А-2 | subRequestId | UUID | Да | UUID подзапроса, выделенного из базового, SQL-выражение которого передается Витрине |
| А-3 | estimatedSize | Long | Да | Оценка объема результата выполнения запроса, в байтах, заполняется только для isForEstimation=true |
| А-4 | estimatedRowCount | Long | Да | Оценка количества строк результата выполнения запроса, заполняется только для isForEstimation=true |
| А-5 | estimatedTime | Long | Да | Оценка времени выполнения запроса в миллисекундах, заполняется только для isForEstimation=true. |

###### Ответ на запрос в случае ошибки (мнемоника Витрины>.query.err)

Таблица 20 – Структура передачи ответа на запрос в случае ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "subRequestId": UUID  "replyTo": String  "errorCode": String  "message": String  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | UUID базового запроса, пришедшего в ПОДД |
| А-2 | subRequestId | UUID | Да | UUID подзапроса, выделенного из базового, SQL-выражение которого передается Витрине |
| А-3 | replyTo | String | Да | Служебное поле. Формируется из запроса |
| А-4 | errorCode | String | Да | Код ошибки (коды должны быть приведены в документации на Витрину) |
| А-5 | message | String | Да | Сообщение об ошибке |

##### Получение БЛОБ по ссылке

###### Запрос на получение БЛОБ по ссылке от Витрины (<мнемоника Витрины>.blob.rq)

Таблица 21 – Структура передачи запроса на получение БЛОБ по ссылке от Витрины

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID,  "queryRequestId": UUID,  "priority": QueryPriority,  "reference": {  "subRequestId": UUID  "path": String  }  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | UUID запроса на выгрузку БЛОБ |
| А-2 | queryRequestId | UUID | Да | Идентификатор исходного запроса, в рамках которого была получена ссылка |
| А-3 | priority | QueryPriority | Да | Приоритет запроса, enumeration: [ NORMAL | HIGH ] |
| А-4 | reference | object | Да | Информация о ссылке |
| А-4.1 | subRequestId | UUID | Да | Идентификатор подзапроса, в рамках которого сформирована ссылка |
| А-4.2 | path | String | Да | Ссылка на выгрузку БЛОБ, полученная от Витрины |

###### Ответ на запрос -БЛОБ (<мнемоника Витрины>.blob.rs)

Таблица 22 – Структура передачи ответа на запрос БЛОБ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID,  "queryRequestId": UUID  "chunkNum": Integer,  "isLast": Boolean  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | UUID запроса |
| А-2 | queryRequestId | UUID | Да | Идентификатор исходного запроса, в рамках которого была получена ссылка |
| А-3 | chunkNum | Integer | Да | Номер чанка |
| А-4 | isLast | Boolean | Да | Признак последнего чанка |

###### Ответ на запрос - уведомление об ошибке (<мнемоника Витрины>.blob.err)

Таблица 23 – Структура передачи ответа на запрос БЛОБ – уведомление об ошибке

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID,  "queryRequestId": UUID  "errorCode": String,  "errorMessage": String  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | UUID запроса |
| А-2 | queryRequestId | UUID | Да | Идентификатор исходного запроса, в рамках которого была получена ссылка |
| А-3 | errorCode | String | Да | Код ошибки |
| А-4 | message | String | Да | Сообщение об ошибке |

##### Отмена запроса

###### Запрос на отмену запроса (<мнемоника Витрины>.cancel.rq)

Таблица 24 – Структура передачи отмены запроса

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | UUID запроса, который нужно отменить |

###### Ответ на запрос на отмену - уведомление об успешности (<мнемоника Витрины>.cancel.rs)

Таблица 25– Структура передачи запроса на отмену – уведомление об успешной отмене

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "isSuccess": Boolean  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | UUID запроса, который нужно отменить |
| А-2 | isSuccess | Boolean | Да | Признак отмены запроса |

###### Ответ на запрос на отмену - уведомление об ошибке (<мнемоника Витрины>.cancel.err)

Таблица 26– Структура передачи запроса на отмену – уведомление об ошибке

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "errorCode": String  "message": String  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | UUID запроса, который нужно отменить |
| А-2 | errorCode | String | Да | Код ошибки |
| А-3 | message | String | Да | Сообщение об ошибке |

#### Информационный обмен с использованием подписок

##### Регистрация подписки на уведомления

###### Запрос на регистрацию подписки к Витрине (топик – <мнемоника Витрины>.replication.rq)

Таблица 27 – Структура передачи запроса данных на регистрацию подписки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "subscriptionId": UUID  "datamartMnemonic": String  "sql": String  "isReplication": Boolean  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | Уникальный идентификатор запроса на регистрацию подписки уведомления |
| А-2 | subscriptionId | UUID | Да | Идентификатор подписки уведомления |
| А-3 | datamartMnemonic | String | Да | Мнемоника Витрины Поставщика данных |
| А-4 | sql | String | Да | Набор SELECT SQL-выражений, отделенных друг от друга точкой с запятой.  (SQL соответствует стандарту ansi 2003, диалект ПОДД) |
| А-5 | isReplication | Boolean | Да | Признак типа подписки:   * false - подписка на уведомления |

###### Ответ на запрос на регистрацию подписки от Витрины - структура данных (топик – <мнемоника Витрины>.replication.rs)

Таблица 28 – Структура передачи ответа на запрос при регистрации подписки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "subscriptionId": UUID  "datamartMnemonic": String  "lastSynId": Integer  "tables" : Array  [  {  "tableId" : UUID  "sql" : String  "fields": Array  [  {  "name": String  "type": String  "length": Integer  "precision": Integer  "scale": Integer  "primaryKey": Integer  "shardingKey": Integer  }  ]  }  ]  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | Уникальный идентификатор запроса на регистрацию подписки на уведомления |
| А-2 | subscriptionId | UUID | Да | Идентификатор подписки на уведомления |
| А-3 | datamartMnemonic | String | Да | Мнемоника Витрины Поставщика данных |
| А-4 | lastSynId | Integer | Да | Последнее значение ID дельты |
| А-6 | lastSynId | Integer | Да | Номер первой дельты, которая будет выгружена в рамках этой подписки |
| А-5 | tables | Array of object | Да | Массив таблиц, которые должны быть созданы для хранения данных подписки |
| А-5.1 | tableId | UUID | Да | UUID таблицы |
| А-5.2 | sql | String | Да | SQL-запрос из списка присланных в запросе, для хранения данных которого требуется эта таблица |
| А-5.3 | fields | Array of object | Да | Массив описаний полей этих таблиц |
| А-5.3.1 | name | String | Да | Имя поля |
| А-5.3.2 | type | String | Да | Тип данных поля |
| А-5.3.3 | length | Integer | Нет | Длина поля |
| А-5.3.4 | precision | Integer | Нет | Точность поля, общее количество знаков в целой и дробной части |
| А-5.3.5 | scale | Integer | Нет | Количество цифр в дробной части, null или 0 означает что это целое число |
| А-5.3.6 | primaryKey | Integer | Нет | Порядковый номер поля (начиная с 0) в составе первичного ключа (null – если не входит в состав первичного ключа) |
| А-5.3.7 | shardingKey | Integer | Нет | Порядковый номер поля (начиная с 0) в составе ключа шардирования (null – если не входит в состав ключа шардирования). |

###### Ответ на запрос на регистрацию подписки в случае ошибки (топик – <мнемоника Витрины>.replication.err)

Таблица 29 – Структура передачи ответа на запрос при регистрации подписки в случае ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "subscriptionId": UUID  "errorCode": String  "message": String  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | Уникальный идентификатор запроса на регистрацию подписки на уведомления |
| А-2 | subscriptionId | UUID | Да | Идентификатор подписки на уведомления |
| А-3 | errorCode | String | Да | Код ошибки (коды должны быть приведены в документации на Витрину) |
| А-4 | message | String | Да | Сообщение об ошибке |

##### Получение дельты

###### Запрос на получение дельты к Витрине (топик – <мнемоника Витрины>.delta.rq)

Таблица 30 – Структура передачи запроса данных на получение дельты

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "subscriptionId": UUID  "fromId": Integer  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | Уникальный идентификатор запроса |
| А-2 | subscriptionId | UUID | Да | Идентификатор подписки на уведомления |
| А-7 | fromId | Integer | Нет | Минимальный номер дельты, начиная с которой требуется выгрузить пакет дельт. Если содержит null, то это первый запрос дельт по этой подписке |

###### Ответ на запрос на получение дельты к Витрине - дельта (топик – <мнемоника Витрины>.delta.rs)

Таблица 31 – Структура передачи ответа на запрос дельты

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "subscriptionId": UUID  "sql": String  "synId": Integer  "minSynId": Integer  "maxSynId": Integer  "synTime": Long  "streamNumber": Integer  "streamTotal": Integer  "isLastChunk": Integer  "chunkNumber": Integer  "replicaHash": String  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | Уникальный идентификатор запроса |
| А-2 | subscriptionId | UUID | Да | Идентификатор подписки на уведомления |
| А-3 | sql | String | Да | Один из SELECT SQL-выражений из запроса на регистрацию |
| А-13 | synId | Integer | Да | Номер дельты, к которой относится фрагмент этого сообщения |
| А-14 | minSynId | Integer | Да | Номер первой передаваемой в этом пакете дельты |
| А-15 | maxSynId | Integer | Да | Номер последней передаваемой в этом пакете дельты |
| А-16 | synTime | Long | Да | UTC дата и время дельты с точностью до секунд относительно Unix epoch (целое количество секунд относительно 1970-01-01 00:00:00) |
| А-6 | streamNumber | Integer | Нет | Номер потока (нумерация начинается с 1) |
| А-7 | streamTotal | Integer | Нет | Количество потоков в этой (synId) дельте |
| А-8 | isLastChunk | Integer | Да | Признак последнего чанка |
| А-9 | chunkNumber | Integer | Да | Номер чанка |
|  |  |  |  |  |

В теле сообщения передается чанк дельты, сериализованной в avro с динамической схемой данных дельты.

Дополнительно к колонкам, присутствующим в запросе, добавляется колонка sys\_op (0 – добавление/изменение записи, 1 – удаление записи).

###### Ответ на запрос на получение дельты к Витрине в случае ошибки (топик – <мнемоника Витрины>.delta.err)

Таблица 32 – Структура передачи ответа на запрос дельты в случае ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "subscriptionId": UUID  "errorCode": String  "message": String  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | Уникальный идентификатор запроса |
| А-2 | subscriptionId | UUID | Да | Идентификатор подписки на уведомления |
| А-3 | errorCode | String | Да | Код ошибки (коды должны быть приведены в документации на Витрину) |
| А-4 | message | String | Да | Сообщение об ошибке |

##### Получение уведомлений о новых данных по подписке

###### Уведомление о событии об изменении данных (топик – <мнемоника Витрины>.delta.notification)

Таблица 33 – Структура уведомления об изменении данных в Витрине

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "subscriptions": Array  [  {  "subscriptionId": UUID  "fromId": Integer  }  ]  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | Уникальный идентификатор запроса |
| А-3 | subscriptions | Array of object | Да |  |
| А-3.1 | subscriptionId | UUID | Да | Идентификатор подписки на уведомления, для которой есть новые данные |
| А-3.4 | fromId | Integer | Да | Номер дельты, начиная с которой есть новые данные |

##### Отмена подписки

###### Запрос на отмену подписки (топик – <мнемоника Витрины>.replication.cancel.rq)

Таблица 34 – Структура запроса данных на отмену подписки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "subscriptionId": UUID  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | UUID запроса |
| А-2 | subscriptionId | UUID | Да | UUID подписки на уведомления |

###### Ответ на запрос на отмену подписки (топик – <мнемоника Витрины>.replication.cancel.rs)

Таблица 35 – Структура ответа на запрос по отмене подписки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "subscriptionId": UUID  "success": Boolean  "message": String  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | UUID запроса |
| А-2 | subscriptionId | UUID | Да | UUID подписки |
| А-3 | success | Boolean | Да | Признак успешной отмены |
| А-4 | message | String | Нет | Описание проблемы в случае ошибки |

#### Получение статистики по Витринам

##### Запрос на получение статистики (топик – <мнемоника Витрины>.statistic.rq)

Таблица 36 – Структура запроса данных на получение статистики

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "protocol": String  "requestId": UUID  "datamart":  {  "mnemonic": String  "version":  {  "major": Integer  "minor": Integer  }  }  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | protocol | String | Да | Не используется |
| А-2 | requestId | UUID | Да | UUID запроса на получение статистики |
| А-3 | datamart | object | Да | Витрина, для которой нужно вернуть статистику |
| А-3.1 | mnemonic | String | Да | Мнемоника Витрины |
| А-3.2 | version | object | Да | Версия |
| А-3.2.1 | major | Integer | Да |  |
| А-3.2.2 | minor | Integer | Да |  |

##### Ответ на запрос на получение статистики - статистика (топик – <мнемоника Витрины>.statistic.rs)

Таблица 37 – Структура ответа на запрос – статистика

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "protocol": String  "requestId": UUID  "datamart":  {  "mnemonic": String  "version":  {  "major": Integer,  "minor": Integer  }  "tables": Array  [  {  "mnemonic": String  "columns": Array  [  {  "mnemonic": String  "notGreater10": Double  "inRange11And100": Double  "inRange101And1000": Double  "moreThan1000": Double  }  ]  }  ]  }  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | protocol | String | Да | Заполняется из запроса |
| А-2 | requestId | UUID | Да | UUID запроса на получение статистики |
| А-3 | datamart | object | Да | Витрина, для которой нужно вернуть статистику |
| А-3.1 | mnemonic | String | Да | Мнемоника Витрины |
| А-3.2 | version | object | Да | Версия |
| А-3.2.1 | major | Integer | Да |  |
| А-3.2.2 | minor | Integer | Да |  |
| А-3.3 | tables | Array of object | Да | Массив таблиц, для которых посчиталась статистика |
| А-3.3.1 | mnemonic | String | Да | Мнемоника класса |
| А-3.3.2 | columns | Array of object | Да | Массив атрибутов (колонок) |
| А-3.3.2.1 | mnemonic | String | Да | Мнемоника атрибута |
| А-3.3.2.2 | notGreater10 | Double | Да | Процент записей с уникальностью меньше или равной 10 |
| А-3.3.2.3 | inRange11And100 | Double | Да | Процент записей с уникальностью [11..100] |
| А-3.3.2.4 | inRange101And1000 | Double | Да | Процент записей с уникальностью [101..1000] |
| А-3.3.2.5 | moreThan1000 | Double | Да | Процент записей с уникальностью больше 1000 |

##### Ответ на запрос статистики в случае ошибки (топик – <мнемоника Витрины>.statistic.err)

Таблица 38 – Структура ответа на запрос статистики в случае ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "protocol": String  "requestId": UUID  "errorCode": String  "message": String  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | protocol | String | Да | Заполняется из запроса |
| А-2 | requestId | UUID | Да | UUID запроса |
| А-3 | errorCode | String | Да | Код ошибки (коды должны быть приведены в документации на Витрину) |
| А-4 | message | String | Да | Сообщение об ошибке |

#### Получение событий от Витрины

##### Уведомление о событии на Витрине (топик – <мнемоника Витрины>.scl.signal)

| {  "eventSCL":  {  "requestId": UUID  "timestampUTC": Int64  "codeEvent": String  "subRequestId": UUID  "datamartId": String  "customerId": String  "hasTableParams": Boolean  }  } | | | | |
| --- | --- | --- | --- | --- |
| Код | Атрибут | Тип | Обязательность | Описание |
| А-1 | eventSCL | object | Да | Событие Витрины для передачи в СЦЛ |
| А-1.1 | requestId | string (UUID) | Да | Уникальный идентификатор исходного запроса Потребителя |
| А-1.2 | timestampUTC | integer (int64) | Да | Таймштамп события, миллисекунды |
| А-1.3 | codeEvent | string | Да | Идентификатор типа события (11015...11021) |
| А-1.4 | subRequestId | string (UUID) | Нет | Уникальный идентификатор подзапроса |
| А-1.5 | queryRequestId | string (UUID) | Нет | requestId родительского запроса, к которому относится данный запрос BLOB по ссылке |
| А-1.6 | datamartId | string | Нет | Мнемоника Витрины данных |
| А-1.7 | customerId | string | Нет | Мнемоника ИС Потребителя |
| А-1.8 | isForEstimation | boolean | Нет | Признак исполнения "требуется оценка" |
| А-1.9 | reportMnemonic | string | Нет | Мнемоника печатной формы |
| А-1.10 | hasTableParams | boolean | Нет | Признак запроса с табличными параметрами |
| А-1.11 | dataVolume | integer (int64) | Нет | Размер ответа в байтах |
| А-1.12 | isMppr | boolean | Нет | Признак исполнения запроса с использованием MPPR |
| А-1.13 | errorCode | string | Нет | Код ошибки |

### Примеры реализации взаимодействия с Агентом ПОДД СМЭВ с использованием брокера сообщений Apache Kafka

Далее приведены примеры программного кода, реализующие следующие процедуры с использованием брокера сообщений Apache Kafka:

* получение запроса от Агента ПОДД СМЭВ;
* передача результата Агенту ПОДД СМЭВ.

Пример проекта для реализации взаимодействия с Агентом приведен в Приложении 2.

#### Получение запроса от Агента ПОДД СМЭВ

Пример программного кода для получения запроса от Агента ПОДД СМЭВ с использованием брокера сообщений Apache Kafka:

|  |
| --- |
| package ru.rtlabs.sample    import mu.KLogging  import org.apache.kafka.clients.consumer.ConsumerRecord  import org.apache.kafka.clients.consumer.KafkaConsumer  import org.apache.kafka.clients.producer.KafkaProducer  import org.apache.kafka.clients.producer.Producer  import ru.rtlabs.sample.messaging.MessageHandler  import ru.rtlabs.sample.query.QueryRequestMessageHandler  import ru.rtlabs.sample.tableParams.TableParamStorageStub  import java.time.Duration  import java.util.Properties    /\*\*  \* Пример конфигурации обработчиков сообщений от агента  \*/  suspend fun main() {    val kafkaProperties = loadProperties("/kafka.properties")  val producer = KafkaProducer<ByteArray, ByteArray>(kafkaProperties)    val topicProperties = loadProperties("/topic.properties")  val handlers = mapOf<String, MessageHandler>(  queryRequestHandler(topicProperties, producer),  )    KafkaConsumer<ByteArray, ByteArray>(kafkaProperties).apply {  subscribe(handlers.keys)  }.use {  while (true) {  it.poll(Duration.ofMillis(1\_000)).forEach { record ->  val topic = record.topic()  val handler = handlers[topic] ?: HandlerNotFound(topic)  handler.handle(record)  }  }  }  }    fun loadProperties(fileName: String) = Properties().apply {  this::class.java.getResourceAsStream(fileName).use {  load(it)  }  }    fun queryRequestHandler(topicProperties: Properties, producer: Producer<ByteArray, ByteArray>) = Pair(  topicProperties.getProperty("query.requestTopicName"),  QueryRequestMessageHandler(  producer = producer,  errorTopic = topicProperties.getProperty("query.errorTopicName"),  resultTopic = topicProperties.getProperty("query.resultTopicName"),  estimationTopic = topicProperties.getProperty("query.estimationTopicName"),  tableParamStorage = TableParamStorageStub(),  ),  )    private class HandlerNotFound(val topic: String) : MessageHandler {  override suspend fun handle(message: ConsumerRecord<ByteArray, ByteArray>) {  logger.error { "Не сконфигурирован обработчик для топика $topic" }  }    companion object : KLogging()  } |

#### Передача результата Агенту ПОДД СМЭВ

Пример программного кода для передачи результата выполнения запрос Агенту ПОДД СМЭВ с использованием брокера сообщений Apache Kafka:

|  |
| --- |
| package ru.rtlabs.sample.query    import org.apache.kafka.clients.producer.Producer  import org.apache.kafka.clients.producer.ProducerRecord  import ru.rtlabs.common.model.DatamartErrorType  import ru.rtlabs.common.model.KeyValueMessage  import ru.rtlabs.common.model.metadata.ColumnInfo  import ru.rtlabs.common.model.metadata.ColumnType  import ru.rtlabs.common.result.resultEncoder  import ru.rtlabs.common.serialization.AvroCodec  import ru.rtlabs.common.serialization.UUIDCodec  import ru.rtlabs.contract.datamart.query.\*  import ru.rtlabs.contract.datamart.query.blob.BinaryReference  import ru.rtlabs.sample.messaging.RequestChunkedResultHandler  import ru.rtlabs.sample.messaging.putHeaders  import ru.rtlabs.sample.tableParams.TableParamStorage  import java.io.ByteArrayOutputStream  import java.math.BigDecimal  import java.nio.ByteBuffer  import java.time.LocalDate  import java.time.LocalDateTime  import java.util.UUID    /\*\*  \* Реализация обработчика входящих сообщений на выполнение  \* sql запросов к базе  \*/  class QueryRequestMessageHandler(  producer: Producer<ByteArray, ByteArray>,  errorTopic: String,  resultTopic: String,  private val estimationTopic: String,  private val tableParamStorage: TableParamStorage,  ) : RequestChunkedResultHandler<  UUID,  DatamartExecuteQueryRequest,  DatamartExecuteQueryResultChunk,  DatamartExecuteQueryError,  >(  producer = producer,  requestKeyDecoder = UUIDCodec::decode,  requestDecoder = AvroCodec(DatamartExecuteQueryRequest.serializer()).decodeFunction(),  resultKeyEncoder = AvroCodec(DatamartExecuteQueryResultChunk.serializer()).encodeFunction(),  resultTopic = resultTopic,  errorKeyEncoder = { UUIDCodec.encode(it.subRequestId) },  errorEncoder = AvroCodec(DatamartExecuteQueryError.serializer()).encodeFunction(),  errorTopic = errorTopic  ) {  override val logger = logger()    private val estimationCodec = AvroCodec(DatamartQueryEstimationResponse.serializer())    override suspend fun process(request: KeyValueMessage<UUID, DatamartExecuteQueryRequest>) {  val executeRequest = request.value  val initialSql = executeRequest.sql  logger.debug { "Входящий sql запрос [$initialSql]" }  val tableParams = executeRequest.tableParams  if (tableParams.isEmpty()) {  if (executeRequest.isForEstimation) {  respondEstimation(request)  } else {  respond(initialSql, request)  }  return  }    try {  waitUntilAllParametersLoaded(tableParams, executeRequest.subRequestId)  val resultSql = tableParamStorage.replaceTableParams(initialSql, tableParams)  logger.debug { "Sql запрос к исполнению с использованием табличных параметров [$resultSql]" }  respond(resultSql, request)  } finally {  deleteAllParameters(tableParams, executeRequest.subRequestId)  }  }    private fun respondEstimation(request: KeyValueMessage<UUID, DatamartExecuteQueryRequest>) {  val executeRequest = request.value  val estimation = DatamartQueryEstimationResponse(  requestId = executeRequest.requestId,  subRequestId = executeRequest.subRequestId,  estimatedRowCount = 100,  estimatedSize = 10\_000,  estimatedTime = 117,  )    producer.send(  ProducerRecord(  estimationTopic,  UUIDCodec.encode(executeRequest.subRequestId),  estimationCodec.encode(estimation),  ).putHeaders(request.headers),  )  }    private fun respond(sql: String, request: KeyValueMessage<UUID, DatamartExecuteQueryRequest>) {  logger.debug { "Выполняется sql запрос [$sql]" }  val streamTotal = 2  val chunkTotal = 3  val executeRequest = request.value  for (stream in 1..streamTotal) {  for (chunk in 1..chunkTotal) {  val resultValue = prepareResult(executeRequest.subRequestId)  val isLastChunk = chunk == chunkTotal  val resultKey = prepareResultKey(  request = executeRequest,  chunkNumber = chunk,  isLastChunk = isLastChunk,  streamNumber = stream,  streamTotal = streamTotal,  uncompressedSize = resultValue.size,  )  sendResult(KeyValueMessage(resultKey, resultValue, request.headers))  }  }  }    private suspend fun waitUntilAllParametersLoaded(tableParams: List<DatamartTableParam>, subRequestId: UUID) {  tableParams.forEach {  tableParamStorage.wait(subRequestId, it)  }  }    private suspend fun deleteAllParameters(tableParams: List<DatamartTableParam>, subRequestId: UUID) {  tableParams.forEach {  tableParamStorage.delete(subRequestId, it)  }  }    private fun prepareResultKey(  request: DatamartExecuteQueryRequest,  chunkNumber: Int,  isLastChunk: Boolean,  streamNumber: Int,  streamTotal: Int,  uncompressedSize: Int,  ) = DatamartExecuteQueryResultChunk(  requestId = request.requestId,  subRequestId = request.subRequestId,  replyTo = request.replyTo,  chunkNumber = chunkNumber,  isLastChunk = isLastChunk,  streamNumber = streamNumber,  streamTotal = streamTotal,  isFragmented = true,  uncompressedSize = uncompressedSize,  )    private fun prepareResult(request: DatamartExecuteQueryRequest): ByteArray {  val metadata = listOf(  ColumnInfo("string\_col", ColumnType.STRING),  ColumnInfo("long\_col", ColumnType.LONG),  ColumnInfo("int\_col", ColumnType.INTEGER),  ColumnInfo("big\_decimal\_col", ColumnType.BIG\_DECIMAL),  ColumnInfo("double\_col", ColumnType.DOUBLE),  ColumnInfo("float\_col", ColumnType.FLOAT),  ColumnInfo("date\_col", ColumnType.DATE),  ColumnInfo("timestamp\_col", ColumnType.TIMESTAMP),  ColumnInfo("bool\_col", ColumnType.BOOLEAN),  ColumnInfo("binary\_col", ColumnType.BINARY)  )    val buf = ByteBuffer.wrap(ByteArray(4) { it.toByte() })  val reference = BinaryReference(  subRequestId = request.subRequestId,  path = "Высокое дерево на плече Подзорной Трубы, направление к С. от С.-С.-В.",  )    @Suppress("RemoveRedundantCallsOfConversionMethods")  val rows: List<Array<Any?>> = listOf(  arrayOf(  "s1",  1L,  1.toInt(),  BigDecimal.TEN.setScale(3) / BigDecimal("3"),  1.2,  1.2f,  LocalDate.parse("2007-12-03"),  LocalDateTime.parse("2007-12-03T10:15:30").plusNanos(123456),  true,  buf  ),  arrayOf(  "s2",  2L,  2.toInt(),  BigDecimal.TEN.setScale(3) / BigDecimal("6"),  2.2,  2.2f,  LocalDate.parse("2007-12-03"),  LocalDateTime.parse("2007-12-03T10:15:30").minusNanos(654321),  true,  reference  ),  arrayOf(null, null, null, null, null, null, null, null, null, null)  )    return ByteArrayOutputStream().apply {  resultEncoder(metadata, this).use { encoder ->  rows.forEach(encoder::append)  }  }.toByteArray()  }    override fun createError(  request: KeyValueMessage<UUID, DatamartExecuteQueryRequest>,  e: Exception,  type: DatamartErrorType?,  ) = DatamartExecuteQueryError(  requestId = request.value.requestId,  subRequestId = request.value.subRequestId,  replyTo = request.value.replyTo,  errorCode = (type ?: DatamartErrorType.QUERY).code,  message = e.message ?: ""  )    } |

## Протокол взаимодействия Агента ПОДД СМЭВ и ИС Потребителя данных

Для обеспечения доступности данных Агент ПОДД СМЭВ предоставляет:

* REST-интерфейс для выполнения запросов к Витринам Поставщиков данных (п. 2.4.1 данного документа);
* специализированный протокол для исполнения запросов с использованием JDBC-интерфейса Агента ПОДД СМЭВ (п. 2.4.1.3 данного документа).

Настройка параметров подключения ИС Потребителя данных осуществляется с помощью конфигурационного файла Агента ПОДД СМЭВ.

Для работы с REST-интерфейсом рекомендуется задать следующие настройки:

* rest-server.host и rest-server.port – параметры, по которым доступен для подключения REST-сервис для отправки тестовых запросов в СМЭВ и управления Агентом ПОДД СМЭВ;
* rest-query-endpoint.host и rest-query-endpoint.port – параметры, по которым доступен для подключения REST-сервис выполнения SQL-запросов;
* async-sql-query-storage-timeout – время хранения (дедлайн) результатов асинхронного запроса в REST-сервисе;
* chunkReorderMaxBufferSize – максимальный размер буфера для сортировки чанков результата исполнения SQL-запроса;
* byteArraySizeLimit – максимальный размер массива для бинарных данных в результате исполнения SQL-запроса.

С целью выполнения запросов к REST-сервису с использованием REST-интерфейса есть возможность задать параметры HTTP-клиента для приёма запросов на стороне Потребителя данных: api-gateway.server.host и api-gateway.server.port.

Для работы с JDBC-драйвером должны быть специфицированы соответствующие хост и порт:

* query-server.host – ip-адрес или хост с которого будет предоставляться доступ к сервису;
* query-server.port – номер порта, на котором Агент ПОДД СМЭВ предоставляет интерфейс для протокола доступа к данным.

Описание конфигурационного файла Агента ПОДД СМЭВ приведено в Руководстве администратора Агента ПОДД СМЭВ[[12]](#footnote-13).

### REST-интерфейс Агента ПОДД СМЭВ

В Агенте ПОДД СМЭВ реализована поддержка REST-интерфейса для выполнения запросов к Витринам Поставщиков данных.

URL-адрес для выполнения обращений к REST-интерфейсу имеет следующий формат: http://<адрес>:<порт>/query, где:

* <адрес> – IP-адрес Агента ПОДД СМЭВ;
* <порт> – порт, на котором развернут REST-интерфейс.

Входные параметры, включая текст SQL-запроса (в случае обращения к Витринам Поставщиков данных), должны кодироваться в виде JSON-строки и передаваться в теле запроса.

Результат выполнения SQL-запроса передается в теле HTTP-ответа в виде JSON-строки. Файлы, передаваемые в составе результата выполнения SQL-запроса, включаются в JSON как строковые атрибуты, кодирующие содержимое передаваемого файла с использованием кодировки Base64.

Возможно выполнение запросов:

* к Витринам данных Поставщиков данных в синхронном режиме (см. п. 2.4.1.1);
* к Витринам данных Поставщиков данных в асинхронном режиме (см. п. 2.4.1.2);
* к REST-сервису ИС Поставщика (см. п. 2.4.1.3).

#### Выполнение SQL-запросов (синхронный режим)

В синхронном режиме получение результата осуществляется путем выполнения   
HTTP-запроса от ИС Потребителя к Агенту ПОДД.

В рамках HTTP-запроса (метод POST) передается SQL-запрос, в ответе возвращается результат выполнения SQL-запроса.

##### HTTP-запрос (метод POST)

Параметры HTTP-запроса (метод POST) приведены в таблице 39.

Таблица 39 – Параметры HTTP-запроса в синхронном режиме

| **№** | **Параметр** | **Тип** | **Обязательность** | **Описание** |
| --- | --- | --- | --- | --- |
| **header** | | | | |
| 1 | Content-Type | string | Да | application/x-www-form-urlencoded; charset=utf-8 |
| 2 | Accept-version | string | Да | Основная (major) часть версии (сейчас 1) |
| **query** | | | | |
|  | async | boolean | Нет | Для синхронного режима выполнения запроса параметр должен отсутствовать или иметь значение False |
| **body** | | | | |
| 1 | priority | string | Да | Приоритет запроса. Варианты:   * NORMAL; * HIGH |
| 2 | timeout | string | Нет | Предельное время ожидания выполнения запроса.  В случае отсутствия параметра в запросе таймаут не выставляется |
| 3 | sql | string | Да | Текст SQL-запроса к Витринам Поставщиков данных |
| 4 | params | array | Нет | Параметры запроса |
| 4.1 | type | string | Да | Тип параметра |
| 4.2 | value | string | Да | Значение параметра |
| 5 | maxrows | string | Нет | Максимальное количество возвращаемых записей таблицы ответа. Если не задан, возвращаются все записи |

Пример запроса:

POST «https://10.81.4.30:29354/query?async=false»

Accept-Version:1

Content-Type:application/x-www-form-urlencoded; encoding=utf-8

priority:NORMAL

timeout:60

sql:SELECT ao.oktmo, o.name, o.kod from fias.addrobj ao LEFT JOIN oktmo.oktmo o on ao.oktmo = o.kod2 WHERE ao.offname= ? AND o.regionid = ?

params:{ “type”: “STRING”, “value”:”Москва”},{ “type”: “INTEGER”, “value”: “18”}

##### Ответ на HTTP-запрос (метод POST)

Допустимые коды возврата:

* 200 – ок;
* 400 – ошибка в запросе, информация об ошибке содержится в параметре «error»;
* 403 – нет полномочий на выполнение запроса, в том числе при блокировке полномочий на стороне Поставщика (с отображением соответствующего текста ошибки);
* 404 – результат по заданному идентификатору SQL-запроса не найден;
* 406 – неподдерживаемая версия протокола (после внедрения поддержки обратной совместимости возвращается только для несуществующих версий);
* 429 – ИС УВ временно заблокирована в связи с превышением лимитов;
* 500 – системная ошибка (в связи в принятыми ограничениями по доступности код 500 предполагается только при сбое Агента);
* 503 – система временно недоступна, возможно повторить запрос через 50 мс.

Параметры ответа с кодом возврата 200 приведены в таблице 40.

Таблица 40 – Параметры ответа с кодом возврата 200

| **№** | **Параметр** | **Тип** | **Обязательность** | **Описание** |
| --- | --- | --- | --- | --- |
|  | responseCode | numeric | Да | Код возврата (HTTP-код) |
| **header** | | | | |
|  | Content-Type | string | Да | application/vnd.ru.rtlabs.podd.agent+json; version=1.0; charset=utf-8 |
| **body** | | | | |
| 1 | created\_at | dateTime | Да | Время формирования ответа |
| 2 | query\_id | string | Да | Идентификатор запроса |
| 3 | rows | array | Нет | Массив записей таблицы ответа (в случае если результат выполнения запроса в виде таблицы). При задании параметра «maxrows» ограничивается его значением |
| 3.1 |  | array | Да | Массив значений. Возможным значением может быть содержимое файла, закодированное в формате BASE64 |
| 4 | meta | array | Да | Список полей результата |
| 4.1 | name | string | Да | Имя поля |
| 4.2 | type | string | Да | Тип поля |

Пример ответа с кодом возврата 200:

HTTP/1.1 200 OK

{

“created\_at”: “2017-12-15T07:36:-03Z”,

“query\_id”: “c005a0e7-0d26-4ce0-a1fa-10c8bdf4dfc5”,

      “meta”: [

      {

           “name”: “count”,

           “type”: “INTEGER”

      }

],

«rows»: [

        [

              «4994»

        ]

]

}

Параметры ответа с кодом возврата, отличным от 200, приведены в таблице 41.

Таблица 41 – Параметры ответа с кодом возврата, отличным от 200

| **№** | **Параметр** | **Тип** | **Обязательность** | **Описание** |
| --- | --- | --- | --- | --- |
|  | responseCode | numeric | Да | Код возврата (HTTP-код) |
| **header** | | | | |
| 1 | Content-Type | string | Да | application/vnd.ru.rtlabs.podd.agent+json; version=1.0; charset=utf-8 |
| **body** | | | | |
| 1 | query\_id | string | Нет | Идентификатор запроса |
| 2 | created\_at | dateTime | Нет | Время формирования ответа |
| 3 | error | string | Нет | Текст ошибки |

Пример ответа с кодом возврата, отличным от 200:

|  |
| --- |
| HTTP/1.1 429 Too Many Requests    {  “created\_at”: “2021-09-10T15:23:36Z”,  “query\_id”: “1ec124b1-1aa6-66d6-9d40-f55438428f56”,  “error”: “RuntimeException: Ошибка во входящем потоке : CustomRSocketException: LIMIT\_EXCEEDED: Запросы к Ядру ПОДД временно заблокированы до September 10, 2021 3:24:35 PM UTC, код причины блокировки=2, подробности: ‘Превышен лимит по количеству запросов lockId=fe462d49-3c5a-4350-8bf6-da155225c52f, userId=e92e3fd4-28d9-48e5-8079-e377b676c9b4 reqCountLimit=10, QueriesStatistic(totalSent=10, totalBytes=2870, totalRows=0, requestIds={1ec124b0-bce0-613b-9d40-c7c6585b14bc=287B, 1ec124b0-caa5-677c-9d40-ebd5d2c0409b=287B, 1ec124b0-d464-695d-9d40-bbafa86207b7=287B, 1ec124b0-dcd5-63ae-9d40-171562fc3a47=287B, 1ec124b0-e493-6a6f-9d40-174dab9e6065=287B, 1ec124b0-ec15-60a0-9d40-f300de6b4e34=287B, 1ec124b0-f6de-6451-9d40-51926f0b5d81=287B, 1ec124b0-fe95-65e2-9d40-e5e27f87babb=287B, 1ec124b1-0581-6d43-9d40-6301af351da7=287B, 1ec124b1-0c9c-6ad4-9d40-dfc04d23e0bc=287B})’.”  } |

#### Выполнение SQL-запросов (асинхронный режим)

В асинхронном режиме получение результата осуществляется путем выполнения двух HTTP-запросов от ИС Потребителя к Агенту ПОДД СМЭВ:

1. В рамках первого HTTP-запроса (метод POST) передается SQL-запрос, в ответе возвращается идентификатор запроса.

Примечание. Для отмены выполнения SQL-запроса необходимо выполнить   
HTTP-запрос с использованием метода DELETE с указанием ID отменяемого запроса.

1. В рамках второго HTTP-запроса (метод GET) передается ранее полученный идентификатор запроса, в ответе возвращается результат выполнения SQL-запроса.

Получение результата по указанному идентификатору возможно только один раз.

##### HTTP-запрос передачи SQL-запроса в Агент ПОДД СМЭВ

###### HTTP-запрос (метод POST)

Параметры HTTP-запроса передачи SQL-запроса в Агент ПОДД СМЭВ приведены в таблице 42.

Таблица 42 – Параметры HTTP-запроса передачи SQL-запроса в Агент ПОДД СМЭВ

| № | Параметр | Тип | Обязательность | Описание |
| --- | --- | --- | --- | --- |
| **header** | | | | |
| 1 | Content-Type | string | Да | application/x-www-form-urlencoded; charset=utf-8 |
| 2 | Accept-version | string | Да | Основная (major) часть версии (сейчас 1) |
| **query** | | | | |
|  | async | boolean | Нет | Для асинхронного режима выполнения запроса  параметр должен иметь значение True |
| **body** | | | | |
| 1 | priority | string | Да | Приоритет запроса. Варианты:   * NORMAL; * HIGH |
| 2 | timeout | string | Нет | Предельное время ожидания выполнения запроса.  В случае отсутствия параметра в запросе таймаут не выставляется |
| 3 | sql | string | Да | Текст SQL-запроса |
| 4 | params | array | Нет | Параметры запроса |
| 4.1 | type | string | Да | Тип параметра |
| 4.2 | value | string | Да | Значение параметра |
| 5 | maxrows | string | Нет | Максимальное количество возвращаемых записей таблицы ответа. Если не задан, возвращаются все записи |

Пример запроса:

POST «https://10.81.4.30:29354/query?async=true»

Accept-Version:1

Content-Type:application/x-www-form-urlencoded; encoding=utf-8

priority:NORMAL

timeout:60

sql:SELECT ao.oktmo, o.name, o.kod from fias.addrobj ao LEFT JOIN oktmo.oktmo o on ao.oktmo = o.kod2 WHERE ao.offname= ? AND o.regionid = ?

params:{ “type”: “STRING”, “value”:”Москва”},{ “type”: “INTEGER”, “value”: “18”}

###### HTTP-запрос с табличным параметром (метод POST)

Параметры HTTP-запроса передачи SQL-запроса с табличным параметром в Агент ПОДД СМЭВ приведены в таблице 43.

Таблица 43 – Параметры HTTP-запроса передачи SQL-запроса с табличным параметром в Агент ПОДД СМЭВ

| № | Параметр | Тип | Обязательность | Описание |
| --- | --- | --- | --- | --- |
| **query** | | | | |
| 1 | async | boolean | Нет | Для асинхронного режима выполнения запроса  параметр должен иметь значение True |
| **header** | | | | |
| 1 | Content-Type | string | Да | multipart/form-data; |
| 2 | Connection | string | Да | keep-alive |
| 3 | Keep-Alive: | string | Да | 300 |
| **boundary--** | | | | |
| 1 | Content-Disposition |  |  | application/json name=request |
| 2 | Content-Type | string | Да | application/json; charset=utf-8 |
| 3 | Accept-version | string | Да | Основная (major) часть версии (сейчас 1) |
| **json** | | | | |
| 1 | priority | string | Да | Приоритет запроса Варианты:   * NORMAL; * HIGH |
| 2 | timeout | string | Нет | Предельное время ожидания выполнения запроса.  В случае отсутствия параметра в запросе таймаут не выставляется |
| 3 | sql | string | Да | Текст SQL-запроса |
| 4 | params | array | Нет | Параметры запроса |
| 4.1 | type | string | Да | Тип параметра |
| 4.2 | value | string | Да | Значение параметра |
| **--boundary--** | | | | |
| 1 | Content-Disposition | string | Да | form-data; name=”table1”; filename=”table1.csv” |
| **file** | | | | |
|  |  |  | Да | Файл в формате csv |

###### Ответ на HTTP-запрос (метод POST)

Допустимые коды возврата:

* 201 – запрос создан;
* 400 – ошибка в запросе, информация об ошибке содержится в параметре «error»;
* 403 – нет полномочий на выполнение запроса, в том числе при блокировке полномочий на стороне Поставщика (с отображением соответствующего текста ошибки);
* 429 – ИС УВ временно заблокирована в связи с превышением лимитов;
* 503 – система временно недоступна, возможно повторить запрос через 50 мс.

Параметры ответа с кодом возврата 201 приведены в таблице 44.

Таблица 44 – Параметры ответа с кодом возврата 201

| № | Параметр | Тип | Обязательность | Описание |
| --- | --- | --- | --- | --- |
|  | responseCode | numeric | Да | Код возврата (HTTP-код) |
| **header** | | | | |
| 1 | Content-Type | string | Да | application/vnd.ru.rtlabs.podd.agent+json; version=1.0; charset=utf-8 |
| 2 | Location | string | Да | Временная ссылка на скачивание результата выполнения запроса |
| **body** | | | | |
| 1 | id | string | Да | Уникальный идентификатор SQL-запроса |
| 2 | deadline | string | Да | Время, до которого доступен результат выполнения запроса |

Пример ответа с кодом возврата 201:

HTTP/1.1 201 Created

{

“id”: “a2f05175-d5bc-47d4-9b88-17930630683e”,

«deadline»: «2021-05-13T06:33:43Z»

}

Параметры ответа с кодом возврата 400 приведены в таблице 45.

Таблица 45 – Параметры ответа с кодом возврата 400

| Параметр | Тип | Обязательность | Описание |
| --- | --- | --- | --- |
| responseCode | numeric | Да | Код возврата (HTTP-код) |
| **header** | | | |
| Content-Type | string | Да | application/vnd.ru.rtlabs.podd.agent+json; version=1.0; charset=utf-8 |
| **body** | | | |
| error | string | Нет | Текст ошибки |

##### HTTP-запрос получения результата ранее переданного в Агент ПОДД СМЭВ асинхронного SQL-запроса

###### HTTP-запрос (метод GET)

Параметры HTTP-запроса получения результата ранее переданного в Агент ПОДД СМЭВ  
SQL-запроса приведены в таблице 46.

Таблица 46 – Параметры HTTP-запроса получения результата ранее переданного в Агент ПОДД СМЭВ SQL-запроса

| Параметр | Тип | Обязательность | Описание |
| --- | --- | --- | --- |
| **header** | | | |
| Accept | string | Да | application/vnd.ru.rtlabs.podd.agent+json; charset=utf-8 |
| Accept-version | string | Да | Основная (major) часть версии (сейчас 1) |
| **query** | | | |
| query\_id | string | Да | Уникальный идентификатор SQL-запроса |

Пример запроса:

GET “https://10.81.4.30:29354/query/c005a0e7-0d26-4ce0-a1fa-10c8bdf4dfc5”

###### Ответ на HTTP-запрос (метод GET)

Допустимые коды возврата:

* 503 – система временно недоступна, возможно повторить запрос через 50 мс;
* 500 – системная ошибка (в связи в принятыми ограничениями по доступности код 500 предполагается только при сбое Агента);
* 429 – ИС УВ временно заблокирована в связи с превышением лимитов;
* 406 – неподдерживаемая версия протокола (после внедрения поддержки обратной совместимости возвращается только для несуществующих версий);
* 404 – Результат по заданному идентификатору SQL-запроса не найден;
* 403 – нет полномочий на выполнение запроса, в том числе при блокировке полномочий на стороне Поставщика (с отображением соответствующего текста ошибки);
* 400 – ошибка в запросе, информация об ошибке содержится в параметре «error»;
* 202 – результат по заданному идентификатору SQL-запроса еще не поступил;
* 200 – ок.

Параметры ответа с кодом возврата 200 приведены в таблице 47.

Таблица 47 – Параметры ответа с кодом возврата 200

| № | Параметр | Тип | Обязательность | Описание |
| --- | --- | --- | --- | --- |
|  | responseCode | numeric | Да | Код возврата (HTTP-код) |
| **header** | | | | |
|  | Content-Type | string | Да | application/vnd.ru.rtlabs.podd.agent+json; version=1.0; charset=utf-8 |
| **body** | | | | |
| 1 | created\_at | dateTime | Да | Время формирования ответа. Время, с которого ответ доступен для получения по запросу |
| 2 | query\_id | string | Да | Идентификатор запроса |
| 3 | rows | array | Нет | Массив записей таблицы ответа (в случае если результат выполнения запроса таблица). При задании параметра «maxrows» ограничивается его значением |
| 3.1 |  | array | Да | Массив значений. Возможным значением может быть содержимое файла закодированное в формате BASE64 |
| 4 | meta | array | Да | Cписок полей результата |
| 4.1 | name | string | Да | Имя поля |
| 4.2 | type | string | Да | Тип поля |

Пример ответа с кодом возврата 200:

HTTP/1.1 200 OK

{

“created\_at”: “2017-12-15T07:36:03Z”,

“query\_id”: “c005a0e7-0d26-4ce0-a1fa-10c8bdf4dfc5”,

“meta”: [

{

“name”: “count”,

“type”: “INTEGER”

}

],

«rows»: [

[

«4994»

]

]

}

Параметры ответа с кодом возврата, отличным от 200, приведены в таблице 48.

Таблица 48 – Параметры ответа с кодом возврата, отличным от 200

| **№** | **Параметр** | **Тип** | **Обязательность** | **Описание** |
| --- | --- | --- | --- | --- |
|  | responseCode | numeric | Да | Код возврата (HTTP-код) |
| **header** | | | | |
|  | Content-Type | string | Да | application/vnd.ru.rtlabs.podd.agent+json; version=1.0; charset=utf-8 |
| **body** | | | | |
| 1 | query\_id | string | Нет | Идентификатор запроса |
| 2 | created\_at | dateTime | Нет | Время формирования ответа – время, с которого ответ доступен для получения по запросу |
| 3 | error | string | Нет | Текст ошибки |

##### HTTP-запрос отмены выполнения ранее переданного в Агент ПОДД СМЭВ асинхронного SQL-запроса

###### HTTP-запрос (метод DELETE)

Параметры HTTP-запроса отмены выполнения ранее переданного в Агент ПОДД СМЭВ  
SQL-запроса приведены в таблице 49.

Таблица 49 – Параметры HTTP-запроса отмены выполнения SQL-запроса

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Параметр** | **Тип** | **Обязательность** | **Описание** |
| **header** | | | | |
| 1 | Accept | string | Да | application/vnd.ru.rtlabs.podd.agent+json; charset=utf-8 |
| 2 | Accept-version | string | Да | Основная (major) часть версии (сейчас 1) |
| **query** | | | | |
|  | query\_id | string | Да | Уникальный идентификатор SQL-запроса |

Пример запроса:

DELETE “https://10.81.4.30:29354/query/c005a0e7-0d26-4ce0-a1fa-10c8bdf4dfc5”

###### Ответ на HTTP-запрос (метод DELETE)

Допустимые коды возврата:

* 200 – ок;
* 400 – ошибка в запросе, информация об ошибке содержится в параметре «error»;
* 404 – результат по заданному идентификатору SQL-запроса не найден;
* 406 – неподдерживаемая версия протокола (после внедрения поддержки обратной совместимости возвращается только для несуществующих версий);
* 500 – системная ошибка (в связи в принятыми ограничениями по доступности код 500 предполагается только при сбое Агента);
* 503 – система временно недоступна, возможно повторить запрос через 50 мс.

Параметры ответа с кодом возврата 200 приведены в таблице 50.

Таблица 50 – Параметры ответа с кодом возврата 200

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Параметр | Тип | Обязательность | Описание |
|  | responseCode | numeric | Да | Код возврата (HTTP-код) |
| **header** | | | | |
|  | Content-Type | string | Да | application/vnd.ru.rtlabs.podd.agent+json; version=1.0; charset=utf-8 |
| **body** | | | | |
| 1 | query\_id | string | Да | Уникальный идентификатор SQL-запроса |
| 2 | success | string | Да | Строка с константным значением «Запрос отменён» |

Пример ответа с кодом возврата 200:

HTTP/1.1 200 OK

{

“id”: “c005a0e7-0d26-4ce0-a1fa-10c8bdf4dfc5”,

«success»: «Запрос удалён»

}

Параметры ответа с кодом возврата, отличным от 200, приведены в таблице 51.

Таблица 51 – Параметры ответа с кодом возврата, отличным от 200

| № | Параметр | Тип | Обязательность | Описание |
| --- | --- | --- | --- | --- |
|  | responseCode | numeric | Да | Код возврата (HTTP-код) |
| **header** | | | | |
|  | Content-Type | string | Да | application/vnd.ru.rtlabs.podd.agent+json; version=1.0; charset=utf-8 |
| **body** | | | | |
| 1 | query\_id | string | Нет | Идентификатор запроса |
| 2 | created\_at | dateTime | Нет | Время формирования ответа – время, с которого ответ доступен для получения по запросу |
| 3 | error | string | Нет | Текст ошибки |

#### Выполнение запросов к REST-сервису ИС Поставщика

Все запросы выполняются в синхронном режиме в соответствии с загруженной в ПОДД СМЭВ спецификацией OpenApi REST-сервиса ИС Поставщика. Примеры спецификаций OpenAPI REST-сервисов ИС Поставщиков приведены в п. 1.6.4 данного документа.

Получение результата осуществляется путем выполнения HTTP-запроса от ИС Потребителя данных к Агенту Поставщика данных. При выполнении запроса могут быть использованы все типы HTTP-методов (POST, GET, PUT и DELETE), выполняющих CRUD-операции (Create/Read/Update/Delete), которые используются в спецификациях OpenAPI, описывающих REST-сервисы ИС Поставщиков.

Ответ на запрос включает статус выполнения операции и (в зависимости от запрошенного метода) тело ответа, содержащее запрошенные данные и/или дополнительную информацию о результате выполнения операции, в соответствии с загруженной в ПОДД спецификацией OpenAPI REST-сервиса ИС Поставщика.

### JDBC-интерфейс Агента ПОДД СМЭВ

Агент ПОДД поддерживает специализированный протокол для исполнения запросов, эталонная реализация которого представлена JDBC-драйвером.

Настройка JDBC-драйвера осуществляется с помощью передачи специализированной адресной строки: protocol://hostname:port/, где:

* protocol – протокол взаимодействия – значение всегда будет «podd»;
* hostname – имя сервера или его IP-адрес;
* port – порт, на котором Агент Потребителя данных предоставляет интерфейс для работы протокола взаимодействия.

#### Пример использования JDBC-драйвера в «Kotlin»

Для прикладного разработчика работа с JDBC драйвером PODD ничем не отличается от работы с обычным JDBC драйвером. Особенность только в URL, которым инициализируется драйвер.

package dev.nsud.jdbc

import org.junit.jupiter.api.Test

import java.sql.Connection

import java.sql.DriverManager

import org.junit.jupiter.api.Assertions

import java.sql.SQLException

class Features {

private getConnectionURI() {

val host = System.getProperty("agent.host", "localhost")

val port = System.getProperty("agent.port", "8182")

return "jdbc:podd://$host:$port"

}

@Test

fun `ожидается успешное соединение с базой данных`() {

Assertions.assertDoesNotThrow { DriverManager.getConnection(getConnectionURI()) }

}

@Test

fun `ожидается успешное исполнение запроса вида "select 1" `() {

Assertions.assertDoesNotThrow {

val con = DriverManager.getConnection(getConnectionURI())

val statement = con.createStatement()

statement.queryTimeout = 5 // таймаут на выполнение запроса - 5 секунд

   statement.setMaxRows(100) // ограничение выборки по кол-ву возвращаемых строк

statement.executeQuery("select 1")

val resultSet = statement.resultSet

Assertions.assertEquals(1, resultSet.getInt(0))

}

}

@Test

fun `ожидается ошибка при исполнении запроса "select 1" `() {

Assertions.assertDoesNotThrow {

val con = DriverManager.getConnection(getConnectionURI())

val statement = con.createStatement()

try {

statement.executeQuery("select 1")

} catch (e: SQLException) {

// получение кода ошибки

Assertions.assertEquals(17089, e.errorCode)

Assertions.assertEquals("PODD-17089: Ошибка при обработке запроса", e.message)

}

}

}

@Test

fun `ожидается успешное получение бинарных данных `() {

Assertions.assertDoesNotThrow {

val expect = getExpectedBytes()

val con = DriverManager.getConnection(getConnectionURI())

val statement = con.createStatement()

statement.executeQuery("select binaryColumn from datamart.table where id=1")

val resultSet = statement.resultSet

Assertions.assertEquals(expect, resultSet.getBlob(0))

}

}

@Test

fun `ожидается успешное применение табличных параметров `() {

Assertions.assertDoesNotThrow {

DriverManager.getConnection(getConnectionURI()).use { connection ->

connection.prepareStatement("select \* from @p1, @p2, oktmo.oktmo o where @p1.a = @p2.b and o.id = @p1.a").use { ps ->

ps.queryTimeout = 10 // таймаут на выполнение запроса - 10 секунд

               ps.setMaxRows(100) // ограничение выборки по кол-ву возвращаемых строк

ps as PoddPreparedStatement

ps.addTableParam(

"p1",

listOf(

ColumnInfo("a", ColumnType.INTEGER),

ColumnInfo("av", ColumnType.STRING),

),

iterator<Array<Any?>> {

yield(arrayOf(1, "1\_1"))

yield(arrayOf(2, "1\_2"))

yield(arrayOf(3, "1\_3"))

},

)

ps.addTableParam(

"p2",

listOf(

ColumnInfo("b", ColumnType.INTEGER),

ColumnInfo("bv", ColumnType.STRING),

),

iterator<Array<Any?>> {

yield(arrayOf(1, "2\_1"))

yield(arrayOf(2, "2\_2"))

yield(arrayOf(3, "2\_3"))

},

)

ps.executeQuery().use { rs ->

(1..rs.metaData.columnCount).forEach {

println("${rs.metaData.getColumnName(it)}: ${rs.metaData.getColumnTypeName(it)}")

}

rs.readFully().forEach { row ->

println()

row.forEach {

print("$it\t")

}

}

}

}

}

}

}

#### Коды возврата

Допустимые коды возврата при ошибках выполнения запроса приведены в таблице 52.

Таблица 52 – Коды возврата при ошибках выполнения запроса

| № | Код возврата | Описание ошибки |
| --- | --- | --- |
|  | 17001 | Внутренняя ошибка |
|  | 17473 | Запрос не прошел проверку корректности (соответствие синтаксису) |
|  | 17471 | ИС УВ временно заблокирована в связи с превышением лимитов |
|  | 17800 | Запрос содержит указание на неподдерживаемую Витрину |
|  | 17472 | Нет полномочий на выполнение запроса |
|  | 17510 | Запрос отменен Потребителем |
|  | 17520 | Запрос отменен по таймауту |
|  | 17404 | Выполнение запроса прекращено из-за блокировки полномочий по результатам проверки на стороне Поставщика |
|  | 17405 | Выполнение запроса прекращено из-за блокировки полномочий по результатам проверки на стороне Поставщика (заблокировано по умолчанию) |
|  | 17406 | Выполнение запроса прекращено из-за блокировки по результату проверки SQL выражения на стороне Поставщика |

## Протокол взаимодействия Агента ПОДД СМЭВ и Хранилища данных по подписке Потребителя данных

Протокол коммуникации Агента ПОДД СМЭВ и Хранилища данных по подписке, расположенных в контуре Потребителя данных, устроен в виде обмена сообщениями с использованием зарезервированных топиков брокера сообщений Apache Kafka.

Брокер сообщений является частью Агента ПОДД СМЭВ и все взаимодействие между Хранилищем данных по подписке и Агентом ПОДД СМЭВ происходит исключительно с использованием брокера сообщений.

### Перечень топиков брокера сообщений Apache Kafka

Топики брокера сообщений приведены в таблице Таблица 53 (названия топиков приведены в качестве примера и могут быть изменены в конфигурации Агента ПОДД СМЭВ).

Таблица 53 – Названия топиков брокера сообщений Apache Kafka

| № | Топик | Публикатор | Подписчик | Передаваемый объект |
| --- | --- | --- | --- | --- |
|  | <мнемоника Витрины>.replication.in.rq | Агент | Хранилище | Структура таблиц Витрины Поставщика данных |
|  | <мнемоника Витрины>  replication.in.rs | Хранилище | Агент | Уведомление об успешном создании структуры данных |
|  | <мнемоника Витрины>  replication.in.err | Хранилище | Агент | Уведомление об ошибке при создании структуры данных |
|  | <мнемоника Витрины>.delta.in.rq | Агент | Хранилище | Пакет дельт |
|  | <мнемоника Витрины>  delta.in.rs | Хранилище | Агент | Уведомления об успешном применении дельт из пакета |
|  | delta.in.err | Хранилище | Агент | Уведомление об ошибке при применении дельт из пакета |

### Структуры сообщений для взаимодействия с Потребителем данных

#### Общие требования

Необязательность атрибута означает возможность указания в качестве значения null. Сам атрибут в сообщении должен присутствовать.

#### Информационный обмен с использованием подписок

##### Создание структуры данных

###### Запрос на создание структуры данных (топик – <мнемоника Витрины>.replication.in.rq)

Таблица 54 – Запрос на создание структуры данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "subscriptionId": UUID  "datamartMnemonic": String  "lastSynId": Integer  "tables" : Array  [  {  "tableId" : UUID  "sql" : String  "fields": Array  [  {  "name": String  "type": String  "length": Integer  "precision": Integer  "scale": Integer  "primaryKey": Integer  "shardingKey": Integer  }  ]  }  ]  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | Уникальный идентификатор запроса |
| А-2 | subscriptionId | UUID | Да | Идентификатор подписки на уведомления |
| А-3 | datamartMnemonic | String | Да | Мнемоника Витрины Потребителя данных |
| А-6 | lastSynId | Integer | Да | Номер первой дельты, которая будет выгружена в рамках этой подписки |
| А-5 | tables | Array of object | Да | Массив таблиц, которые должны быть созданы для хранения изменений в данных Поставщиков |
| А-5.1 | tableId | UUID | Да | Идентификатор таблицы |
| А-5.2 | sql | String | Да | Один из запросов |
| А-5.3 | fields | Array of object | Да | Массив описаний полей этих таблиц |
| А-5.3.1 | name | String | Да | Имя поля |
| А-5.3.2 | type | String | Да | Тип поля |
| А-5.3.3 | length | Integer | Нет | Длина поля |
| А-5.3.4 | precision | Integer | Нет | Точность поля, общее количество знаков в целой и дробной части |
| А-5.3.5 | scale | Integer | Нет | Количество цифр в дробной части, null или 0 означает что это целое число |
| А-5.3.6 | primaryKey | Integer | Нет | Порядковый номер поля (начиная с 0) в составе первичного ключа (null – если не входит в состав первичного ключа) |
| А-5.3.7 | shardingKey | Integer | Нет | Порядковый номер поля (начиная с 0) в составе ключа шардирования (null – если не входит в состав ключа шардирования). |

###### Ответ на запрос на создание структуры данных - уведомление об успешности (топик – <мнемоника Витрины>.replication.in.rs)

Таблица 55 – Ответ на запрос на создание структуры данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "subscriptionId": UUID  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | Уникальный идентификатор запроса |
| А-2 | subscriptionId | UUID | Да | Идентификатор подписки на уведомления |

###### Ответ на запрос на создание структуры данных - уведомление об ошибке (топик – <мнемоника Витрины>.replication.in.err)

Таблица 56 – Ответ на запрос на создание структуры данных – уведомление об ошибке

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "subscriptionId": UUID  "errorCode": String  "message": String  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | Уникальный идентификатор запроса |
| А-2 | subscriptionId | UUID | Да | Идентификатор подписки на уведомления |
| А-3 | errorCode | String | Да | Код ошибки (коды должны быть приведены в документации на Витрину) |
| А-4 | message | String | Да | Сообщение об ошибке |

##### Передача дельты

###### Запрос на приём дельты (топик – <мнемоника Витрины>.delta.in.rq)

Таблица 57 – Структура запроса данных на приём дельты

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "subscriptionId": UUID  "sql": String  "synId": Integer  "minSynId": Integer  "maxSynId": Integer  "synTime": Long  "streamNumber": Integer  "streamTotal": Integer  "isLastChunk": Integer  "chunkNumber": Integer  "replicaHash": String  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | Уникальный идентификатор запроса |
| А-2 | subscriptionId | UUID | Да | Идентификатор подписки на уведомления |
| А-3 | sql | String | Да | Один из SELECT SQL-выражений из запроса на регистрацию |
| А-12 | synId | Integer | Да | Номер дельты, к которой относится фрагмент этого сообщения |
| А-13 | minSynId | Integer | Да | Номер первой передаваемой в этом пакете дельты |
| А-14 | maxSynId | Integer | Да | Номер последней передаваемой в этом пакете дельты |
| А-15 | synTime | Long | Да | UTC дата и время дельты с точностью до секунд относительно Unix epoch (целое количество секунд относительно 1970-01-01 00:00:00) |
| А-5 | streamNumber | Integer | Нет | Номер потока (нумерация начинается с 1) |
| А-6 | streamTotal | Integer | Нет | Количество потоков в этой (synId) дельте |
| А-7 | isLastChunk | Integer | Да | Признак последнего чанка |
| А-8 | chunkNumber | Integer | Да | Номер чанка |
|  |  |  |  |  |

В теле сообщения передается чанк дельты, сериализованной в avro с динамической схемой данных дельты.

Дополнительно к колонкам, присутствующим в запросе, добавляется колонка sys\_op (0 – добавление/изменение записи, 1 – удаление записи).

###### Ответ на запрос на приём дельты - уведомление об успешности (топик – <мнемоника Витрины>.delta.in.rs)

Таблица 58 – Структура данных при ответе на запрос получения дельты

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "subscriptionId": UUID  "synId": Integer  "isReplication": Boolean  "isSnapshot": Boolean  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | Уникальный идентификатор запроса |
| А-2 | subscriptionId | UUID | Да | Идентификатор подписки на уведомления |
| А-3 | synId | Integer | Да | Номер последней дельты, принятой в рамках этого пакета дельт |
| А-4 | isReplication | Boolean | Да | Признак типа подписки:   * false – подписка на уведомления |

###### Ответ на запрос на приём дельты - уведомление об ошибке (топик – <мнемоника Витрины>.delta.in.err)

Таблица 59 – Структура данных при ответе на запрос получения дельты – уведомление об ошибке

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| {  "requestId": UUID  "subscriptionId": UUID  "errorCode": String  "message": String  } | | | | |
| **Код** | **Атрибут** | **Тип** | **Обязательность** | **Описание** |
| А-1 | requestId | UUID | Да | Уникальный идентификатор запроса |
| А-2 | subscriptionId | UUID | Да | Идентификатор подписки на уведомления |
| А-5 | synId | Integer | Да | Номер дельты, на которой возникла ошибка |
| А-3 | errorCode | String | Да | Код ошибки (коды должны быть приведены в документации на Витрину) |
| А-4 | message | String | Да | Сообщение об ошибке |

## Протокол взаимодействия Агента ПОДД СМЭВ и ИС Поставщика данных

Взаимодействие Агента ПОДД СМЭВ и REST-сервиса на стороне ИС Поставщика осуществляется в соответствии со спецификацией OpenAPI, описывающей REST-сервис ИС Поставщика и загруженной в ПОДД СМЭВ.

Для использования запросов к REST-сервису ИС Поставщика через ПОДД СМЭВ необходимо произвести настройки, указанные в Руководстве администратора Агента ПОДД СМЭВ[[13]](#footnote-14).

# Использование ПОДД СМЭВ

## SQL-синтаксис

Запросы для передачи в Агент ПОДД СМЭВ должны соответствовать следующему синтаксису. Упоминание «booleanExpression» соответствует п. 3.1.1 настоящего документа.

Statement:

query

prioritySpec

statementList:

statement [ ‘;’ statement ]\* [priority][ ‘;’ ]

query:

values

| WITH withItem [ , withItem ]\* query

| {

select

| selectWithoutFrom

| query UNION [ ALL | DISTINCT ] query

| query EXCEPT [ ALL | DISTINCT ] query

| query MINUS [ ALL | DISTINCT ] query

| query INTERSECT [ ALL | DISTINCT ] query

}

[ ORDER BY orderItem [, orderItem ]\* ]

[ LIMIT [ start, ] { count | ALL } ]

[ OFFSET start { ROW | ROWS } ]

[ FETCH { FIRST | NEXT } [ count ] { ROW | ROWS } ONLY ]

withItem:

name

[ ‘(‘ column [, column ]\* ‘)’ ]

AS ‘(‘ query ‘)’

orderItem:

expression [ ASC | DESC ] [ NULLS FIRST | NULLS LAST ]

select:

SELECT [ ‘/\*+’ hint [, hint]\* ‘\*/’ ] [ STREAM ] [ ALL | DISTINCT ]

{ \* | projectItem [, projectItem ]\* }

FROM tableExpression

[ WHERE booleanExpression ]

[ GROUP BY { groupItem [, groupItem ]\* } ]

[ HAVING booleanExpression ]

[ WINDOW windowName AS windowSpec [, windowName AS windowSpec ]\* ]

selectWithoutFrom:

SELECT [ ALL | DISTINCT ]

{ \* | projectItem [, projectItem ]\* }

projectItem:

expression [ [ AS ] columnAlias ]

| tableAlias . \*

tableExpression:

tableReference [, tableReference ]\*

| tableExpression [ NATURAL ] [ ( LEFT | RIGHT | FULL ) [ OUTER ] ] JOIN tableExpression [ joinCondition ]

| tableExpression CROSS JOIN tableExpression

| tableExpression [ CROSS | OUTER ] APPLY tableExpression

joinCondition:

ON booleanExpression

| USING ‘(‘ column [, column ]\* ‘)’

tableReference:

tablePrimary

[ FOR SYSTEM\_TIME AS OF expression ]

[ matchRecognize ]

[ [ AS ] alias [ ‘(‘ columnAlias [, columnAlias ]\* ‘)’ ] ]

tablePrimary:

tableIdentifier

| tablePrimary [ ‘/\*+’ hint [, hint]\* ‘\*/’ ] [ EXTEND ] ‘(‘ columnDecl [, columnDecl ]\* ‘)’

| [ LATERAL ] ‘(‘ query ‘)’

| UNNEST ‘(‘ expression ‘)’ [ WITH ORDINALITY ]

| [ LATERAL ] TABLE ‘(‘ [ SPECIFIC ] functionName ‘(‘ expression [, expression ]\* ‘)’ ‘)’

tableIdentifier:

[ catalogName . ] datamartMnemonic [ . version ] . ] tableName

| ‘(‘ TABLE [ catalogName . ] datamartMnemonic [ . version ] . ] tableName ‘)’

| @tableParameter

| ‘(‘ TABLE @tableParameter ‘)’

tableName:

datamartTableName |

regulatedQueryName[‘(‘ parameter[, parameter]\*’)’]

parameter:

parameterValue | ‘?’

columnDecl:

column type [ NOT NULL ]

hint:

hintName

| hintName ‘(‘ hintOptions ‘)’

hintOptions:

hintKVOption [, hintKVOption]\*

| optionName [, optionName]\*

| optionValue [, optionValue]\*

hintKVOption:

optionName ‘=’ stringLiteral

| stringLiteral ‘=’ stringLiteral

optionValue:

stringLiteral

| numericLiteral

values:

VALUES expression [, expression ]\*

groupItem:

expression

| ‘(‘ ‘)’

| ‘(‘ expression [, expression ]\* ‘)’

| CUBE ‘(‘ expression [, expression ]\* ‘)’

| ROLLUP ‘(‘ expression [, expression ]\* ‘)’

| GROUPING SETS ‘(‘ groupItem [, groupItem ]\* ‘)’

window:

windowName

| windowSpec

windowSpec:

‘(‘

[ windowName ]

[ ORDER BY orderItem [, orderItem ]\* ]

[ PARTITION BY expression [, expression ]\* ]

[

RANGE numericOrIntervalExpression { PRECEDING | FOLLOWING }

| ROWS numericExpression { PRECEDING | FOLLOWING }

]

‘)’

priority:

priorityValues

| NORMAL

| HIGH

prioritySpec:

[ PRIORITY=priorityValues ]

### Поддерживаемые возможности SQL-синтаксиса

| № | Описание | Пример запроса |
| --- | --- | --- |
| **-** | **Числовые типы данных** | |
| 1 | Типы данных INTEGER и SMALLINT | SELECT CAST(1 AS INT); |
| 2 | Типы данных REAL, DOUBLE PRECISION и FLOAT | SELECT CAST(1 AS FLOAT); |
| 3 | Типы данных DECIMAL и NUMERIC | SELECT CAST(1 AS NUMERIC); |
| 4 | Арифметические операторы | SELECT 10+1, 9-2, 8\*3, 7/2; |
| 5 | Числовые сравнения | SELECT 1 WHERE 1 < 2; |
| 6 | Неявные преобразования между числовыми типами данных | SELECT \* FROM t WHERE int\_column = 1.00; |
| **-** | **Символьные типы данных** | |
| 7 | Тип данных CHARACTER  Длина по умолчанию 30 | SELECT CAST(‘1111111111111111111111111111111111111111111’ AS CHAR); |
| 8 | Тип данных CHARACTER VARYING  Длина по умолчанию 30 | SELECT CAST(‘1111111111111111111111111111111111111111111’ AS VARCHAR); |
| 9 | Символьные строки | SELECT ''; |
| 10 | Функция CHARACTER\_LENGTH  убирает завершающие пробелы из значений CHARACTER перед подсчётом символов | SELECT character\_length(char\_column) FROM t; |
| 11 | Функция OCTET\_LENGTH | SELECT octet\_length(char\_column) FROM t; |
| 12 | Функция SUBSTRING | SELECT substring(char\_column FROM 1 FOR 1) FROM t; |
| 13 | Конкатенация символьных строк | SELECT ‘a’ || ‘b’ FROM t; |
| 14 | Функции UPPER и LOWER | SELECT upper(‘a’),lower(‘B’) FROM t;  SELECT \* FROM t WHERE int\_column > (SELECT DISTINCT (int\_column) FROM t); |
| 15 | Функция TRIM | SELECT trim(‘a ‘) FROM t; |
| 16 | Неявные преобразования между типами символьных строк | SELECT \* FROM t WHERE char\_column > varchar\_column; |
| 17 | Функция POSITION | SELECT position(‘A’ IN char\_column) FROM t; |
| 18 | Сравнения символов | SELECT \* FROM t WHERE char\_column > ‘a’; |
| **-** | **Идентификаторы** | |
| 19 | Идентификаторы с разделителями | SELECT 1 AS «t47»; |
| 20 | Идентификаторы в нижнем регистре | SELECT 1 AS t48; |
| 21 | Завершающее подчёркивание | SELECT 1 AS t49\_; |
| **-** | **Базовое определение запросов** | |
| 22 | SELECT DISTINCT | SELECT DISTINCT int\_column FROM t; |
| 23 | Предложение GROUP BY | SELECT DISTINCT int\_column FROM t GROUP BY int\_column; |
| 24 | GROUP BY может содержать колонки не из <списка выборки> | SELECT DISTINCT char\_column FROM t GROUP BY lower(char\_column); |
| 25 | Элементы списка выборки могут переименовываться | SELECT int\_column AS K FROM t ORDER BY K; |
| 26 | Предложение HAVING | SELECT count(\*) FROM t HAVING count(\*) > 0; |
| 27 | Дополнение \* в списке выборки | SELECT t.\* FROM t; |
| 28 | Корреляционные имена в предложении FROM | SELECT \* FROM t AS K; |
| 29 | Переименование колонок в предложении FROM | SELECT \* FROM t AS x(q, c); |
| **-** | **Базовые предикаты и условия поиска** | |
| 30 | Предикат сравнения | SELECT \* FROM t WHERE 0 = 0; |
| 31 | Предикат BETWEEN | SELECT \* FROM t WHERE ‘ ‘ BETWEEN ‘’ AND ‘’; |
| 32 | Предикат IN со списком значений | SELECT \* FROM t WHERE char\_column IN (‘a’, upper(‘a’)); |
| 33 | Предикат LIKE | SELECT \* FROM t WHERE char\_column LIKE ‘\_’; |
| 34 | Предложение ESCAPE в предикате LIKE | SELECT \* FROM t WHERE ‘abc’ LIKE ‘abcX\_’ ESCAPE ‘X’; |
| 35 | Предикат NULL | SELECT \* FROM t WHERE char\_column IS NOT NULL; |
| 36 | Предикаты количественного сравнения | SELECT \* FROM t WHERE char\_column = ANY (SELECT char\_column FROM t); |
| 37 | Предикат EXISTS | SELECT \* FROM t WHERE NOT EXISTS (SELECT char\_column FROM t); |
| 38 | Подзапросы в предикате сравнения | SELECT \* FROM t WHERE int\_column > (SELECT max (int\_column) FROM t); |
| 39 | Подзапросы в предикате IN | SELECT \* FROM t WHERE char\_column IN (SELECT char\_column FROM t); |
| 40 | Подзапросы в предикате количественного сравнения | SELECT \* FROM t WHERE char\_column >= ALL(SELECT char\_column FROM t); |
| 41 | Коррелирующие подзапросы | SELECT \* FROM t WHERE int\_column = (SELECT int\_column FROM t2 WHERE t2.char\_column = t.char\_column); |
| 42 | Условие поиска | SELECT \* FROM t WHERE 0 <> 0 OR ‘a’ < ‘b’ AND int\_column IS NOT NULL; |
| **-** | **Простые выражения с запросами** | |
| 43 | Табличный оператор UNION DISTINCT | SELECT \* FROM t UNION DISTINCT SELECT \* FROM t; |
| 44 | Табличный оператор UNION ALL | SELECT \* FROM t UNION ALL SELECT \* FROM t; |
| 45 | Табличный оператор EXCEPT DISTINCT | SELECT \* FROM t EXCEPT DISTINCT SELECT \* FROM t; |
| 46 | Колонки, объединяемые табличными операторами, могут иметь разные типы данных | SELECT char\_column FROM t UNION SELECT 5; |
| 47 | Табличные операторы в подзапросах | SELECT \* FROM t WHERE ‘a’ IN (SELECT char\_column FROM t UNION SELECT char\_column FROM t); |
| **-** | **Функции множеств** | |
| 48 | AVG | SELECT avg(int\_column) FROM t; |
| 49 | COUNT | SELECT count(int\_column) FROM t; |
| 50 | MAX | SELECT max(int\_column) FROM t; |
| 51 | MIN | SELECT min(int\_column) FROM t; |
| 52 | SUM | SELECT sum(int\_column) FROM t; |
| 53 | Дополнение ALL | SELECT sum(ALL int\_column) FROM t; |
| 54 | Дополнение DISTINCT | SELECT sum(DISTINCT int\_column) FROM t; |
| 55 | **Оператор SELECT, возвращающий одну строку** | SELECT count(\*) FROM t; |
| **-** | **Базовая поддержка курсоров** | |
| 56 | Колонки ORDER BY, отсутствующие в списке выборки | SELECT int\_column FROM t ORDER BY char\_column; |
| 57 | Выражения значений в предложении ORDER BY | SELECT int\_column FROM t ORDER BY -int\_column; |
| 58 | **Поддержка NULL (NULL вместо значений)** | SELECT int\_column FROM t WHERE int\_column IS NULL; |
|  | **Базовое соединение таблиц** | |
| 59 | Внутреннее соединение (но не обязательно с ключевым словом INNER) | SELECT a.int\_column FROM t a JOIN t b ON a.int\_column = b.int\_column; |
| 60 | Ключевое слово INNER | SELECT a.int\_column FROM t a JOIN t b ON a.int\_column = b.int\_column; |
| 61 | LEFT OUTER JOIN | SELECT a.int\_column, b.int\_column FROM t a LEFT OUTER JOIN t b ON a.int\_column = b.int\_column; |
| 62 | RIGHT OUTER JOIN | SELECT a.int\_column, b.int\_column FROM t a RIGHT OUTER JOIN t b ON a.int\_column = b.int\_column; |
| 63 | Внешние соединения могут быть вложенными | SELECT a.int\_column FROM t a LEFT OUTER JOIN t b ON a.int\_column = b.int\_column LEFT OUTER JOIN t2 c ON a.int\_column = c.int\_column; |
| 64 | Внутренняя таблица с левой или правой стороны внешнего соединения может также участвовать во внутреннем соединении | SELECT t2.int\_column FROM (t2 LEFT OUTER JOIN t ON t.int\_column = t2.int\_column) j INNER JOIN t2 ON j.int\_column = t2.int\_column |
| 65 | Поддерживаются все операторы сравнения (а не только =) | SELECT \* FROM t WHERE 0 = 1 OR 0 > 1 OR 0 < 1 OR 0 <> 1; |
| **-** | **Базовая поддержка даты и времени** | |
| 66 | Тип данных DATE (включая поддержку строк DATE) | SELECT ‘2012-07-12’ AS "DATE"; |
| 67 | Тип данных TIME (включая поддержку строк TIME) с точностью до секунд как минимум с 0 знаков после запятой | SELECT ‘1:2:3’ AS "TIME"; |
| 68 | Тип данных TIMESTAMP (включая поддержку строк TIMESTAMP) с точностью до секунд как минимум с 0 и 6 знаками после запятой | SELECT ‘2012-07-12’ AS "TIMESTAMP"; |
| 69 | Предикаты сравнения с типами данных DATE, TIME и TIMESTAMP | SELECT \* FROM t3 WHERE date\_column = date\_column AND time\_column = time\_column AND timestamp\_column = timestamp\_column; |
| 70 | Явное приведение (CAST) между типами даты/времени и типами символьных строк | SELECT cast(date\_column AS VARCHAR(10)) FROM t3; |
| 71 | CURRENT\_DATE | SELECT current\_date FROM t; |
| 72 | LOCALTIME | SELECT localtime FROM t; |
| 73 | LOCALTIMESTAMP | SELECT localtimestamp FROM t; |
| 74 | FOR SYSTEM\_TIME (запрос данных, актуальных на указанную дату и время) | SELECT \* FROM t FOR SYSTEM\_TIME AS OF TIMESTAMP 'YYYY-MM-DD hh:mm:ss' |
| 75 | **Функция CAST** | SELECT cast(int\_column AS INT) FROM t; |
| **-** | **Выражение CASE** | |
| 76 | Простой оператор CASE | SELECT CASE WHEN 1 = 0 THEN 5 ELSE 7 END FROM t; |
| 77 | Оператор CASE с условиями | SELECT CASE 1 WHEN 0 THEN 5 ELSE 7 END FROM t; |
| 78 | NULLIF | SELECT nullif(int\_column, 7) FROM t; |
| 79 | COALESCE | SELECT coalesce(int\_column,7) FROM t; |
| 80 | **Длинные идентификаторы** | SELECT 1 AS A12345678901234567890123456789; |
| 81 | **Спецсимволы Unicode в идентификаторах** | SELECT 1 AS Я12345678901234567890123456789; |
| 82 | **Спецсимволы Unicode в текстовых строках** | SELECT U&'\6553'; |
| 83 | **Национальные символы** | SELECT 'Я'; |
| 84 | **Скалярные значения подзапросов** | SELECT int\_column FROM t WHERE int\_column = (SELECT count(\*) FROM t); |
| 85 | **Расширенный предикат NULL** | SELECT \* FROM t WHERE row(int\_column, int\_column) IS NOT NULL; |

## Примеры запросов с использованием SQL-синтаксиса, поддерживаемого ПОДД СМЭВ

Необходимо учитывать, что возможность выполнения соответствующих синтаксических конструкций зависит от их поддержки со стороны ПО Витрины данных.

### Запрос с использованием табличных выражений

**with** cte **as** (

**select** shortname, regioncode, oktmo **from** fias**.**addrobj

**where** formalname = ‘Москва’

)

**select** \* **from** cte

### Запросы с использованием базовых операторов

1. Запрос с использованием EXCEPT:

**select** oktmo **from** fias**.**addrobj

**EXCEPT**

**select** kod2 **from** oktmo**.**oktmo

1. Запрос с использованием INTERSECT:

**select** oktmo **from** fias**.**addrobj

**INTERSECT**

**select** kod2 **from** oktmo**.**oktmo

1. Запросы с использованием UNION:

**select** oktmo **from** fias.addrobj

**UNION**

**select** kod2 **from** oktmo.oktmo

**select** oktmo **from** fias.addrobj

**UNION** **ALL**

**select** kod2 **from** oktmo.oktmo

1. Запросы с использованием ORDER BY:

**select** shortname, formalname **from** fias.addrobj

**where** aolevel = **1**

**order** **by** formalname

**select** shortname, formalname **from** fias.addrobj

**where** aolevel = **1**

**order** **by** formalname **desc**

1. Запрос использованием GROUP BY, HAVING:

**select** ao.regioncode, **count**(\*) N **from** fias.addrobj ao

**left** **join** oktmo.oktmo o **on** o.kod2 = ao.oktmo

**where** o.kod2 **is** **null**

**group** **by** ao.regioncode

**having** **COUNT**(\*) > **3000**

**order** **by** N **desc**

### Запросы с указанием версии Витрины

**select** \* **from** egrul.**1.0**.legalentity

Результат содержит 4 атрибута в соответствии со структурой таблицы "legalentity" Витрины "egrul" версии 1.0.

**select** \* **from** egrul.**2.0**.legalentity

Результат содержит 5 атрибутов в соответствии со структурой таблицы "legalentity" Витрины "egrul" версии 2.0.

### Запросы с использованием JOIN

**select** le.short\_name, rc.region\_full\_name **from** egrul.legalentity le

**join** egrul.region\_codes rc **on** le.region\_code = rc.region\_code

**SELECT** ao.oktmo, o.name, o.kod

**from** fias.addrobj ao

**JOIN** oktmo.oktmo o **on** ao.oktmo = o.kod2

**WHERE** ao.offname = ‘Москва’ **AND** o.regionname = ‘город Москва’

**select** le.short\_name, le.region\_code, o.regionid, o.name, o.regionname **from** egrul.legalentity le

**left** **join** oktmo.oktmo o **on** **cast**(o.regionid **as** **varchar**) = le.region\_code

**where** o.kod2 **is** **null**

**select** le.short\_name, le.region\_code, o.regionid, o.name, o.regionname **from** egrul.legalentity le

**full join** oktmo.oktmo o **on** **cast**(o.regionid **as** **varchar**) = le.region\_code

**where** le.ogrn **is** **null** **or** o.kod2 **is** **null**

**select** le.short\_name, le.region\_code, o.regionid, o.name, o.regionname **from** egrul.legalentity le

**right** **join** oktmo.oktmo o **on** **cast**(o.regionid **as** **varchar**) = le.region\_code

**where** le.ogrn **is** **null**

**select** le.short\_name, rc.region\_full\_name **from** egrul.legalentity le

**cross** **join** egrul.region\_codes rc

**where** rc.region\_code = le.region\_code

### Регламентированные запросы

1. РЗ в формате параметризированного:

// Исходный запрос от Потребителя данных ПОДД СМЭВ:

select \* from oktmo.1.0.oktmo\_view(‘Московская область’,7)

// Пример преобразования исходного запроса в соответствии с загруженным определением РЗ:

select id, whenadd, name, regionname, settlementtypename FROM oktmo.1.0.oktmo where regionname = ‘Московская область’ AND settlementtypeid = 7

РЗ без параметров:

// Исходный запрос от Потребителя данных ПОДД СМЭВ:

select \* from egrul.1.1.legalentity\_view()

// Пример преобразования исходного запроса в соответствии с загруженным определением РЗ:

select \* from egrul.2.legalentity

распределенный РЗ:

// Исходный запрос от Потребителя данных ПОДД СМЭВ:

select \* from podd.1.1.r\_query(‘Москва’, 18)

// Пример преобразования исходного запроса в соответствии с загруженным определением РЗ:

select ao.oktmo, o.name, o.kod from fias.1.0.addrobj ao LEFT JOIN oktmo.1.0.oktmo o on ao.oktmo = o.kod2 WHERE ao.offname= ‘Москва’ AND o.regionid = 18

## Получение двоичных объектов в результатах запроса

ИС Потребителя данных имеет возможность получения в качестве результата выполнения запроса с использованием ПОДД СМЭВ как скалярных параметров (значение которых представляет собой число, строку или дату), так и двоичных объектов (соответствующих файлам, размещенным на Витринах Потребителей данных).

Для обеспечения возможности получения ИС Потребителя данных двоичных объектов необходимо выполнение следующих условий:

* Витрина Поставщика данных поддерживает тип данных «двоичный объект»;
* в метаданных Витрины для соответствующих атрибутов Витрины Поставщика данных установлен тип данных «двоичный объект» (описание типов данных ПОДД СМЭВ приведено в п. 1.6.4).

В случае выполнения запроса, результат которого содержит атрибуты с типом данных «двоичный объект», в составе результата запроса атрибут может принимать одно из следующих возможных значений:

1. непосредственно сам двоичный объект;
2. уникальная ссылка на получение двоичного объекта с Витрины Поставщика данных (см. шаги на рисунке 13).

При запросе ИС Потребителя данных от Агента ПОДД файла по ссылке формируется запрос двоичного объекта к Витрине Поставщика данных. ПОДД СМЭВ передает ИС Потребителя данных двоичный объект (файл), организуя двоичный поток между Витриной Поставщика данных и ИС Потребителя данных.

В процессе передачи Агент Поставщика данных разбивает поток данных на сегменты (чанки) и подписывает каждый сегмент подписью Поставщика данных. Перед передачей данных Потребителю данных подпись проверяется Ядром ПОДД СМЭВ.

Ниже приведен пример программного кода, реализующего получение двоичного объекта:

**package** dev.nsud.jdbc

**import** **org.junit.jupiter.api.Test**

**import** **java.sql.Connection**

**import** **java.sql.DriverManager**

**import** **org.junit.jupiter.api.Assertions**

**import** **java.sql.SQLException**

**class** **Features** {

**private** **getConnectionURI**() {

**val** host = System.getProperty(“agent.host”, “localhost”)

**val** port = System.getProperty(“agent.port”, “8182”)

**return** “jdbc:podd://$host:$port”

}

**@Test**

**fun** `ожидается успешное соединение с базой данных`() {

Assertions.assertDoesNotThrow { DriverManager.getConnection(getConnectionURI()) }

}

**@Test**

**fun** `ожидается успешное получение бинарных данных `() {

Assertions.assertDoesNotThrow {

val expect = getExpectedBytes()

**val** con = DriverManager.getConnection(getConnectionURI())

**val** statement = con.createStatement()

statement.executeQuery(“select binaryColumn from datamart.table where id=1”)

**val** resultSet = statement.resultSet

Assertions.assertEquals(expect, resultSet.getBlob(**0**))

}

}

}

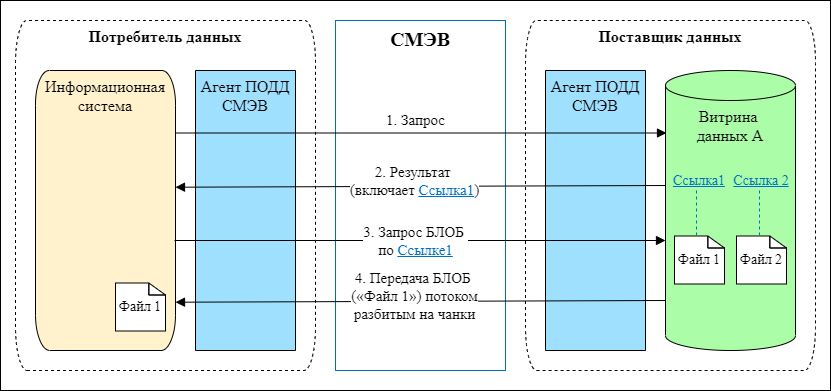


Рисунок 13 – Получение БЛОБ в качестве результата запроса

## Формирование и передача статистики атрибутов Витрины данных Агенту ПОДД СМЭВ

Статистика по атрибутам, получаемая Ядром ПОДД от Витрин данных, используется в алгоритме оптимизации плана выполнения запросов Потребителя данных в целях уменьшения трафика между УВ и Ядром ПОДД.

Для атрибутов таблиц Витрины данных, по которым необходимо сформировать статистику, вычисляются следующие показатели:

1. Показатель «от 0 до 10 повторений» – содержит процент от общего числа строк, для которых повторение значений атрибута не превышает 10.
2. Показатель «от 11 до 100 повторений» – содержит процент от общего числа строк, для которых повторение значений атрибута более 10, но не превышает 100.
3. Показатель «от 101 до 1000 повторений» – содержит процент от общего числа строк, для которых повторение значений атрибута более 100, но не превышает 1000.
4. Показатель «более 1000 повторений» – содержит процент от общего числа строк, для которых повторение значений атрибута более 1000.

Пример статистики по атрибуту приведен в таблице 60.

Таблица 60 – Пример статистики по атрибуту Витрины данных

|  |  |
| --- | --- |
| **Показатель** | **Значение** |
| от 0 до 10 повторений | 98,00 |
| от 11 до 100 повторений | 0,25 |
| от 101 до 1000 повторений | 1,50 |
| более 1000 повторений | 0,25 |

Ниже приведен пример SQL-скрипта по расчету показателей статистики для атрибута <<table>>.<<attribute>>:

**select**

**sum**(**case** **when** z **between** **1** **and** **10** **then** z **else** **0** end) \* **100** / **sum**(z))::**int** z10,

-- показатель «от 0 до 10 повторений»

(**sum**(**case** **when** z **between** **11** **and** **100** **then** z **else** **0** end) \* **100** / **sum**(z))::**int** z100,

-- показатель «от 11 до 100 повторений»

(**sum**(**case** **when** z **between** **101** **and** **1000** **then** z **else** **0** end) \* **100** / **sum**(z))::**int** z1000,

-- показатель «от 101 до 1000 повторений»

(**sum**(**case** **when** z > **1001** **then** z **else** **0** end) \* **100** / **sum**(z))::**int** z\_oth

-- показатель «более 1000 повторений»

**from** (

**select** **count**(\*) z

**from** **<<table>>**

**group** **by** **<<attribute>>**

)t

Расчет показателей статистики осуществляется для атрибутов, по которым создан индекс в Витрине данных.

Рассчитанные показатели статистики по атрибутам Витрины данных передаются Агенту ПОДД с использованием топика «statistics.rs» брокера сообщений Kafka в ответ на полученный запрос статистики с использованием топика «statistics.rq» (см. п. 2.3.2). При этом предполагается соответствие рассчитанных показателей статистики и параметров ответа, приведенное в таблице 61.

Таблица 61 – Соответствие показателей параметрам ответа на запрос статистики

| **Показатель** | **Параметр ответа** |
| --- | --- |
| от 0 до 10 повторений | «notGreater10» |
| от 11 до 100 повторений | «inRange11And100» |
| от 101 до 1000 повторений | «inRange101And1000» |
| более 1000 повторений | «moreThan1000» |

## Выполнение регламентированных запросов

РЗ выполняется аналогично выполнению любого запроса к Витрине Поставщика данных. Описание синтаксиса выполнения запроса приведено в п. 3.1 настоящего документа.

Эта операция может быть выполнена:

* с использованием REST-интерфейса в соответствии с п. 2.4.1 настоящего документа.
* с использованием JDBC-интерфейса в соответствии с п. 2.4.1.3 настоящего документа.

Для выполнения необходимо составить SQL выражение, указав мнемонику РЗ вместо таблицы. Формат РЗ имеет вид:

select \* from <мнемоника Витрины>.<версия РЗ>.<мнемоника РЗ>(<параметры>)

, где:

* <мнемоника Витрины> – для простых запросов (к одной Витрине) задается мнемоника соответствующей Витрины. Для распределенного запроса вместо мнемоники Витрины указывается префикс «podd»;
* <версия РЗ> – задается в формате «major.minor»;
* <мнемоника РЗ> – мнемоника вызываемого РЗ;
* <параметры> – если РЗ является параметризованным, то все параметры должны быть указаны в скобках, через запятую, в порядке их указания в определении РЗ (загруженном из ФГИС «ЕИП НСУД»).

Если выполняемый запрос не требует указания параметров, то скобки должны быть указаны пустыми.

Если параметры не указаны в исходном запросе, но их значения по умолчанию присутствуют в загруженном определении, то они будут подставлены автоматически.

Преобразование запросов и подзапросов при выполнении РЗ:

1. Исходный запрос от Потребителя данных передается в Ядро ПОДД СМЭВ в неизменном виде.
2. В Ядре ПОДД СМЭВ осуществляется обработка запроса: валидация, проверка полномочий, разбиение на подзапросы.
3. Подзапрос, сформированный в Ядре ПОДД СМЭВ к Витрине данных, представляет собой SQL-выражение в соответствии с определением РЗ.

## Выполнение запросов с использованием табличных параметров, передаваемых Потребителем данных для обогащения

Выполнение запроса, включающего табличный параметр, осуществляется аналогично выполнению любого запроса к Витрине Поставщика данных. Описание синтаксиса выполнения запроса приведено в п. 3.1 настоящего документа.

Для выполнения запроса с табличным параметром необходимо:

1. Определить структуру таблицы, указанной в табличном параметре.
2. Указать запрос, содержащий табличный параметр.
3. Указать источник данных для табличного параметра.

Эти операции могут быть выполнены с использованием REST или JDBC-интерфейса.

* с использованием : для описания передаваемой в качестве параметра таблицы и добавления данных расширяются функции JDBC-драйвера. В этом случае

### Запрос с использованием REST-интерфейса

При использовании REST-интерфейса данные собираются в CSV файл, который прикрепляется к запросу, содержащему информацию о структуре таблицы и SQL-выражение. Формат запроса приведен в п. 2.4.1.2 данного документа.

В теле запроса передаются следующие параметры:

sql:SELECT el.inn, er.region\_name FROM @inns el LEFT JOIN egrul.2.region\_codes er ON SUBSTRING(el.inn,1,2) = er.region\_code

priority:NORMAL

tableParams:{“name”: “inns”, “columns”:[{“name”: “id”, “type”: “INTEGER”},{“name”: “inn”, “type”: “STRING”}]}

inns:<l.csv>

, где:

1. sql – текст произвольного SQL-запроса, содержащего табличные параметры;
2. tableParams – описание передаваемого файла с данными для табличного параметра, где:
   * name – табличный параметр;
   * columns – перечень названий столбцов и их типов, содержащихся в файле с данными для табличного параметра.
3. inns – файл с данными для табличного параметра, где:
   * inns – табличный параметр (выступает в качестве названия параметра запроса);
   * <1.csv> – файл в формате csv (поддерживаемый формат), передаваемый в параметре запроса.

Пример csv файла (разделитель – запятая):

1, 4345310593

2, 4311003795

3, 4345336320

### Запрос с использованием JDBC-интерфейса

При использовании JDBC-интерфейса для описание передаваемой в качестве параметра таблицы и добавления данных расширяются функции драйвера JDBC. Для доступа к данным формируется итератор.

После передачи данных выполняется запрос, включающий табличный параметр в формате @имя\_параметра.

Пример использования представлен в п. 2.4.2.1 данного документа.

## Выполнение запросов к REST-сервису ИС Поставщика

Для выполнения запроса к REST-сервису ИС Поставщика необходимо:

1. Определить доступные REST-сервисы.
2. Составить URI запроса в соответствии со спецификацией OpenAPI зарегистрированного REST-сервиса ИС Поставщика, загруженной в ПОДД.

URI запроса формируется путём конкатенации мнемоники Агента Поставщика данных, префикса в URL (basePath) соответствующего REST-сервиса ИС Поставщика и path операции. Сформированный запрос должен совпадать с запросом из спецификации OpenAPI.

Формат запроса для обмена с использованиям REST-сервиса ИС Поставщика имеет вид:

|  |
| --- |
| <HTTP-метод> <адрес>:<порт>/<systemMnemonic><basePath><paths\> |

, где:

* HTTP-метод – метод из поддерживаемых REST-сервисом;
* <адрес> – IP-адрес Агента Потребителя;
* <порт> – порт, на котором развернут REST-интерфейс Агента Потребителя;
* systemMnemonic – мнемоника Агента Поставщика данных;
* basePath – префикс в URL соответствующего REST-сервиса ИС Поставщика;
* paths\ – путь операции, указанный в спецификации OpenAPI соответствующего REST-сервиса ИС Поставщика.

Пример URI запроса соответствует примеру спецификации OpenAPI, приведенному в п. 1.6.4 данного документа:

|  |
| --- |
| GET 10.81.4.30:29164/agent-oktmo/region-service/region/123 |

Выполнение запросов осуществляется через REST-интерфейс Агента ПОДД СМЭВ (см. п. 2.4.1 данного документа).

Приложение 1. Термины и сокращения

| **Сокращение** | **Термин** | **Определение** |
| --- | --- | --- |
| - | *Агент (ПОДД СМЭВ)* | Типовое программное обеспечение, устанавливаемое в контуре ИС УВ и обеспечивающее сопряжение Витрин данных, ИС УВ с ПОДД СМЭВ |
| - | *Актуальная версия* | Последняя среди действующих версия сущности (Витрины данных, РЗ) |
| - | *Аналитический запрос* | Запрос, предполагающий обработку значительного числа строк в Витрине Поставщика данных ПОДД СМЭВ и вычисление агрегатных функций или сравнение массивов данных |
| - | *Бинарный протокол* | Протокол, в котором для передачи данных используются нетекстовые символы. |
| БЛОБ (BLOB) | *(Большой) Двоичный объект* | Тип данных, значение которого представляет собой массив байт, размер которого существенно превышает размер базовых скалярных типов (int, float, double, date) |
| - | *Витрина (данных)*  *реже: Экземпляр ПО "Витрина данных"* | Комплекс программных и технических средств в составе информационно-телекоммуникационной инфраструктуры Участника взаимодействия, обеспечивающий хранение и предоставление данных другим Участникам взаимодействия с использованием ПОДД СМЭВ |
| ВС | *Вид сведения СМЭВ* | Комплекс документальных и программных компонентов, зарегистрированный в СМЭВ 3.х, обеспечивающий взаимодействие ИС ведомств в определённом формате и по определённым правилам |
| - | *Действующая версия* | Версия сущности (Витрины данных, РЗ), для которой срок начала поддержки наступил, а срок окончания поддержки или не наступил, или не задан |
| - |  |  |
| ЕИП НСУД (ФГИС «ЕИП НСУД») | *Федеральная государственная информационная система «Единая информационная платформа национальной системы управления данными»* | - |
| - | *Запрос данных,*  *SQL-запрос* | Произвольный или регламентированный запрос к данным, сформулированный на языке SQL |
| - | *Запрос к REST-сервису ИС Поставщика* | Запрос, сформированный в соответствии со спецификацией OpenAPI, которая зарегистрирована в ПОДД и содержит описание REST-сервиса ИС Поставщика |
| ИС | *Информационная система* | - |
| ИС УВ | *Информационная система Участника взаимодействия* | - |
| ИЭП | *Инфраструктура электронного правительства* | - |
| КЦР | *Конструктор цифровых регламентов* | - |
| ЛК УВ | *Личный кабинет участника взаимодействия* | Система, предназначенная для выполнения регистрационных действий в СМЭВ3.х, декларирования типов данных и атрибутов, проектирования видов сведений, использующихся при межведомственном электронном взаимодействии |
| - | *Мнемоника Витрины* | Уникальное строковое значение, однозначно определяющее Витину данных |
| - | *Мнемоника (символическая) регламентированного запроса* | Уникальное строковое значение, однозначно определяющее регламентированный запрос.  Является частью полной мнемоники РЗ |
|  | *Мнемоника (символическая) таблицы* | Уникальное строковое значение, однозначно определяющее таблицу данных в рамках Витрины данных |
| - | *Многомерный запрос* | Запрос, результат которого размечен так, что все возвращаемые атрибуты поделены на измерения и факты, таким образом многомерный запрос подходит для использования в средствах OLAP-анализа как источник данных |
| МГД | *Модель государственных данных* | Совокупность Моделей данных Витрин и полномочий Потребителей данных ПОДД СМЭВ, загруженных в Ядро ПОДД СМЭВ |
| - | *Модель данных (Витрины)* | Описание структуры Витрины, содержащее перечень сущностей, включая их атрибутный состав и вспомогательные свойства (мнемоника, версия и т.д.) Витрины, загруженные в Ядро ПОДД СМЭВ |
| НСУД | *Национальная система управления данными* | - |
| НФАП | *Национальный фонд алгоритмов и программ* | - |
| ОВ | *Орган власти* | - |
| ОГРН | *Основной Государственный Регистрационный Номер* | - |
| - | *Ответ (на запрос или подзапрос)* | Набор данных, формируемый Поставщиком или Ядром ПОДД СМЭВ в ответ на Запрос Потребителя или Подзапрос Ядра ПОДД СМЭВ |
| ОЭ | *Опытная эксплуатация* | - |
| - | *Параметр запроса* | Символическое имя, входящее в текст SQL-запроса и не содержащееся в Модели данных, в терминах которой сформулирован SQL-запрос.  Значение параметра должно быть определено перед выполнением SQL-запроса.  Значение параметра может быть только данными и не может быть инструкцией языка SQL |
| - | *Параметризованный запрос* | Запрос, содержащий в своём определении Параметр(ы), значение которых Потребитель данных ПОДД задаёт непосредственно перед выполнением запроса |
| ПО | *Программное обеспечение* | - |
| ПОДД СМЭВ  (СМЭВ 4) | *Подсистема обеспечения доступа к данным СМЭВ* | Часть транспортной подсистемы СМЭВ, обеспечивающая доступ к данным, размещённым на Витринах данных или в ИС Поставщика данных |
| - | *Подзапрос (данных)* | SQL-запрос, который отправляется Ядром ПОДД СМЭВ в адрес Поставщика данных |
| - | *Подписка (потребителя)* | Подписка Потребителя данных ПОДД СМЭВ на уведомления об изменении данных в ПОДД СМЭВ |
| - | *Полная мнемоника регламентированного запроса* | Мнемоника сформированная по правилу <мнемоника витрины>.<мнемоника РЗ>.  Если запрос распределенный, то формируется по правилу podd.<мнемоника РЗ> |
| - | *Поставщик данных (ПОДД СМЭВ)* | Участник взаимодействия, являющийся источником данных для других участников и использующий ПОДД СМЭВ для передачи данных |
| - | *Потребитель данных (ПОДД СМЭВ)* | Участник взаимодействия, получающий данные от Поставщиков данных для дальнейшей их обработки и использующий для передачи запросов и получения данных ПОДД СМЭВ |
| - | *Произвольный SQL-запрос* | Запрос со структурой, задаваемой Потребителем данных на языке SQL |
| - | *Простой запрос* | Запрос, содержащий атрибуты только одной Витрины данных |
| - | *Протокол ПОДД СМЭВ* | Бинарный протокол Apache Pulsar с предварительной аутентификацией |
| - | *Распределенный запрос* | Регламентированный запрос, инициированный Потребителем, SQL-выражение которого содержит наборы данных из двух или более Витрин данных |
| РЗ | *Регламентированный SQL-запрос* | SQL-запрос, выраженный в терминах Модели данных, загруженной в ПОДД, и зарегистрированный в Ядре ПОДД под символической мнемоникой, используемой ИС Потребителя ПОДД для выполнения регламентированного запроса.  Может иметь параметры, значения которых задаются Потребителем данных ПОДД при выполнении регламентированного запроса |
| - | *Результат запроса* | Набор данных, полученный в ходе обработки SQL-запроса |
| - |  |  |
| - | *Сегментирование (данных)* | Разделение массива данных на части (сегменты) для возможности независимой обработки каждого сегмента узлами вычислительного кластера в составе Ядра ПОДД СМЭВ |
| - | *Скалярный параметр (запроса)* | Параметр, значение которого представляет собой число, строку или дату |
| - | *Служебное сообщение* | Сообщение, которое содержит служебные данные |
| СМЭВ | *Единая система межведомственного электронного взаимодействия* | - |
| СМЭВ 3 | *Единая система межведомственного электронного взаимодействия, функционирующая в соответствии с Методическими рекомендациям по работе со СМЭВ версии 3.x* | - |
| - |  |  |
| - | *Содержательное сообщение* | Сообщение, которое содержит полезные данные (SQL-запрос и подзапрос, ответы на запрос и подзапрос) и ЭП-ОВ |
| - | *Сообщение* | Набор служебных или полезных данных, передаваемый между Ядром ПОДД СМЭВ и Агентом ПОДД СМЭВ, в виде служебных или содержательных сообщений соответственно |
| СУБД | *Система управления базами данных* | - |
| СЦ | *Ситуационный центр* | ФГИС «Федеральный Ситуационный Центр Электронного Правительства» – инструмент для управления коммуникациями между участниками взаимодействия в ходе процессов создания цифровых регламентов в части шагов, не автоматизированных в КЦР. Доступен по адресу <https://sc.minsvyaz.ru> |
| - | *Табличный параметр (запроса)* | Параметр, значение которого представляет собой двумерный массив с именованными колонками и неупорядоченными строками. Формальный табличный параметр может использоваться в инструкциях FROM, JOIN как источник данных |
| - | *Технологический портал СМЭВ3* | Справочный интернет-ресурс, посвящённый СМЭВ, функционирующей согласно Методическим рекомендациям по работе со СМЭВ версии 3.x; доступен по адресу – <https://smev3.gosuslugi.ru/portal/> |
| УВ | *Участник взаимодействия* | Орган или организация, участвующий в информационном обмене через СМЭВ |
| - | *Универсальные запросы* | Запросы, атрибутный состав которых не имеет разметки, предполагающей какое-то специальное использование этого запроса |
| - | *Фиксированные запросы* | Запросы, не предусматривающие каких-либо возможностей для Потребителя данных ПОДД СМЭВ уточнить запрос |
| - | *Хранилище (данных по подписке)* | Комплекс программных и технических средств в составе информационно-телекоммуникационной инфраструктуры участника взаимодействия, обеспечивающий получение от Агента ПОДД, хранение и предоставление доступа к данным по подписке |
| - | *Хэш-сумма (Хэш-сумма состояния подписки)* | Значение, рассчитанное по данным Витрины, соответствующее подписке по состоянию на момент формирования дельты изменений |
| - | *Чанк* | Фрагмент результирующих данных оптимального для передачи по сети размера |
| ЭП | *Электронная подпись* | - |
| ЭП ОВ | *Электронная подпись органа власти* | - |
| - | *Ядро ПОДД СМЭВ* | Часть ПОДД СМЭВ, предназначенная для проверки и маршрутизации запросов между участниками взаимодействия |
| ACL | *Access Control List* | Список, каждая запись которого определяет субъект воздействия и допустимую операцию |
| API | *Application programming interface* | Программный интерфейс приложения |
| - | *Apache Avro* | линейно-ориентированный (строчный) формат передачи наборов данных, используемый в качестве платформы сериализации, разрабатываемый в рамках фонда [Apache](https://ru.wikipedia.org/wiki/Apache_Software_Foundation) |
| - | *Apache Kafka* | Распределённый программный [брокер сообщений](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%BE%D0%BA%D0%B5%D1%80_%D1%81%D0%BE%D0%BE%D0%B1%D1%89%D0%B5%D0%BD%D0%B8%D0%B9), проект с [открытым исходным кодом](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), разрабатываемый в рамках фонда [Apache](https://ru.wikipedia.org/wiki/Apache_Software_Foundation) |
| - | *Apache Pulsar* | Распределённый программный [брокер сообщений](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%BE%D0%BA%D0%B5%D1%80_%D1%81%D0%BE%D0%BE%D0%B1%D1%89%D0%B5%D0%BD%D0%B8%D0%B9), проект с [открытым исходным кодом](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), разрабатываемый в рамках фонда [Apache](https://ru.wikipedia.org/wiki/Apache_Software_Foundation) |
| CSV | *Comma-Separated Values* | Текстовый формат, предназначенный для представления табличных данных |
| JDBC | *Java DataBase Connectivity* | Стандарт взаимодействия Java-приложений с различными СУБД |
| - | *JDBC-драйвер* | Библиотека классов, реализующая стандарт JDBC и подключения к источнику данных с использованием специализированного протокола, поддерживаемого источником данных |
| - | *JDBC-подключение* | Подключение к источнику данных с помощью JDBC-драйвера |
| JSON | *JavaScript Object Notation* | Общий формат для представления значений и объектов в соответствии со стандартом RFC 4627 |
| - | *Kotlin* | Современный кроссплатформенный язык, совместимый с Java и другими языками программирования.  Разрабатывается Фондом Котлина, в который входят компании Google и JetBrains.  Kotlin – язык со строгой статической типизацией, имеет объектно-ориентированные и функциональный черты, удобен для конструирования DSL. |
| - | *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 |
| protobuf | *Protocol Buffers* | Протокол сериализации структурированных данных |
| REST | *Representational State Transfer – «передача состояния представления»* | Архитектурный стиль взаимодействия компонентов распределённого приложения в сети |
|  | *REST-сервис ИС Поставщика* | Сервис в инфраструктуре Поставщика, обеспечивающий предоставление данных другим Участникам взаимодействия с использованием ПОДД СМЭВ в соответствии с загруженной в ПОДД СМЭВ спецификацией OpenAPI |
| SQL | *Structured Query Language* | Декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных |
| XML | *Extensible markup language* | Расширяемый язык разметки |

Приложение 2. Пример проекта для реализации взаимодействия с Агентом ПОДД СМЭВ

Проект размещен в файле «Приложение 2. Пример проекта для реализации взаимодействия с Агентом ПОДД СМЭВ.zip».

Пример разработан с использованием языка Kotlin, сборка и запуск выполняется стандартным способом для java-проектов – с использованием gradle build tool (<https://gradle.org/>).

1. Описание ВС размещено на технологическом портале СМЭВ3 – <https://smev3.gosuslugi.ru/portal/> [↑](#footnote-ref-2)
2. Перечисленные руководства и регламенты размещены на технологическом портале СМЭВ3 – <https://smev3.gosuslugi.ru/portal/> [↑](#footnote-ref-3)
3. Перечисленные руководства и регламенты размещены на технологическом портале СМЭВ3 – <https://smev3.gosuslugi.ru/portal/> [↑](#footnote-ref-4)
4. Описание ВС размещено на технологическом портале СМЭВ3, <https://smev3.gosuslugi.ru/portal/inquirytype_one.jsp?id=2219248&zone=fed&page=1&dTest=false> [↑](#footnote-ref-5)
5. Описание ВС размещено на технологическом портале СМЭВ3 – <https://smev3.gosuslugi.ru/portal/> [↑](#footnote-ref-6)
6. Размещена на технологическом портале СМЭВ3, <https://smev3.gosuslugi.ru/portal/> [↑](#footnote-ref-7)
7. Размещена на технологическом портале СМЭВ3, <https://smev3.gosuslugi.ru/portal/> [↑](#footnote-ref-8)
8. Руководство администратора Агента ПОДД СМЭВ размещено на технологическом портале СМЭВ3, <https://smev3.gosuslugi.ru/portal/> [↑](#footnote-ref-9)
9. Регламент подключения к СМЭВ 4 размещена на технологическом портале СМЭВ3, <https://smev3.gosuslugi.ru/portal/> [↑](#footnote-ref-10)
10. Далее в разделе 2.3 префикс *<мнемоника Витрины>* в названии топиков может быть опущен. [↑](#footnote-ref-11)
11. Данная информация используется Агентом при работе и обновляется на основе актуальных данных от Ядра при каждом перезапуске. [↑](#footnote-ref-12)
12. Руководство администратора Агента ПОДД СМЭВ размещено на технологическом портале СМЭВ3, <https://smev3.gosuslugi.ru/portal/> [↑](#footnote-ref-13)
13. Руководство администратора Агента ПОДД СМЭВ размещено на технологическом портале СМЭВ3, <https://smev3.gosuslugi.ru/portal/> [↑](#footnote-ref-14)