ИНФРАСТРУКТУРА ЭЛЕКТРОННОГО ПРАВИТЕЛЬСТВА

Методические рекомендации по работе с Подсистемой обеспечения доступа к данным Единой системы межведомственного электронного взаимодействия

Версия 3.9.1

**ПОДД СМЭВ**

Листов 115

Москва, 2023

Содержание

[1 Введение 4](#_Toc153801482)

[1.1 Общие Сведения 4](#_Toc153801483)

[1.1.1 Нормативно-правовые основания 4](#_Toc153801484)

[1.2 Виды информационного обмена с использованием СМЭВ 5](#_Toc153801485)

[1.2.1 Классификация регламентированных SQL-запросов 5](#_Toc153801486)

[1.3 Участники информационного обмена с использованием ПОДД СМЭВ 6](#_Toc153801487)

[1.4 Информационный обмен участников взаимодействия с использованием ПОДД СМЭВ 8](#_Toc153801488)

[1.4.1 Обмен с использованием Регламентированных SQL-запросов 8](#_Toc153801489)

[1.4.2 Обмен с использованием Рассылок 12](#_Toc153801496)

[1.4.3 Обмен с использованием запросов к REST-сервису ИС Ответчика 25](#_Toc153801506)

[1.5 Метаданные ПОДД СМЭВ 34](#_Toc153801516)

[1.5.1 Поддерживаемые форматы 35](#_Toc153801517)

[1.5.2 Модель данных Витрины Поставщика данных 39](#_Toc153801518)

[1.5.3 Подписки Потребителей данных ПОДД СМЭВ на регламентированный запрос типа «Рассылка» 39](#_Toc153801519)

[1.5.4 Определения Регламентированных SQL-запросов 41](#_Toc153801520)

[1.5.5 REST-сервисы ИС Ответчиков 43](#_Toc153801521)

[1.5.6 Полномочия Потребителя данных ПОДД СМЭВ на доступ к данным 50](#_Toc153801522)

[1.6 Типы данных ПОДД СМЭВ 52](#_Toc153801527)

[2 Описание подключения к ПОДД СМЭВ 54](#_Toc153801528)

[2.1 Подключение участников взаимодействия с использованием ПОДД СМЭВ 54](#_Toc153801529)

[2.2 Протокол взаимодействия Агента СМЭВ4 и Витрины Поставщика данных 54](#_Toc153801530)

[2.2.1 Перечень топиков брокера сообщений Apache Kafka 54](#_Toc153801531)

[2.2.2 Настройка Агента СМЭВ4 для работы с несколькими Витринами 56](#_Toc153801532)

[2.2.3 Примеры реализации взаимодействия с Агентом СМЭВ4 с использованием брокера сообщений Apache Kafka 59](#_Toc153801535)

[2.3 Протокол взаимодействия Агента СМЭВ4 и ИС Потребителя данных 64](#_Toc153801538)

[2.3.1 REST-интерфейс Агента СМЭВ4 для SQL-запросов 64](#_Toc153801539)

[2.3.2 REST-интерфейс Агента СМЭВ4 для запросов к REST-сервису ИС Ответчика 78](#_Toc153801560)

[2.3.3 JDBC-интерфейс Агента СМЭВ4 для SQL-запросов 79](#_Toc153801563)

[2.4 Протокол взаимодействия Агента СМЭВ4 и Витрины по подписке Потребителя данных 82](#_Toc153801566)

[2.4.1 Перечень топиков брокера сообщений Apache Kafka 82](#_Toc153801567)

[2.5 Протокол взаимодействия Агента СМЭВ4и ИС Ответчика 83](#_Toc153801568)

[3 Использование ПОДД СМЭВ 84](#_Toc153801569)

[3.1 SQL-синтаксис 84](#_Toc153801570)

[3.2 Примеры запросов с использованием SQL-синтаксиса, поддерживаемого ПОДД СМЭВ 87](#_Toc153801571)

[3.2.1 Запрос с использованием табличных выражений 87](#_Toc153801572)

[3.2.2 Указание в запросе версии Витрины 88](#_Toc153801573)

[3.2.3 Запросы с использованием JOIN 88](#_Toc153801574)

[3.3 Выполнение регламентированных SQL-запросов 88](#_Toc153801575)

[3.3.1 Правила выполнения 88](#_Toc153801576)

[3.3.2 Примеры вызова и преобразования исходного запроса 90](#_Toc153801577)

[3.3.3 Выполнение запроса с системным параметром 92](#_Toc153801578)

[3.3.4 Выполнение запросов с использованием табличных параметров, передаваемых Потребителем данных для обогащения 93](#_Toc153801579)

[3.3.5 Получение двоичных объектов в результатах запроса 94](#_Toc153801582)

[3.3.6 Получение печатной формы в результатах запроса 96](#_Toc153801583)

[3.4 Выполнение запросов к REST-сервису ИС Ответчика 97](#_Toc153801584)

[3.5 Сквозная идентификация запросов 97](#_Toc153801585)

[3.6 Лимитирование регламентированных запросов Потребителя к информационным ресурсам Поставщика 99](#_Toc153801586)

[3.6.1 Лимиты 99](#_Toc153801587)

[3.6.2 Условия для добавления лимитов 100](#_Toc153801588)

[3.6.3 Блокировки 100](#_Toc153801589)

[3.6.4 Управление лимитами и блокировками 101](#_Toc153801590)

[3.6.5 Алгоритм лимитирования 101](#_Toc153801591)

[4 Термины и сокращения 103](#_Toc153801592)

[5 История изменений 109](#_Toc153801593)

# 1 Введение

## 1.1 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 соответствующем разделе.

### 1.1.1 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 Российской Федерации», 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.

## 1.2 Виды информационного обмена с использованием СМЭВ

Виды информационного обмена, поддерживаемые СМЭВ, приведены в [Таблице 1.1](#_13fdc53a3a4f82f82f7b076f0bf69661).

Данный документ описывает рекомендации и требования в отношении 2–4 видов обмена.

Таблица 1.1 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» [[1]](#footnote-1) |
| 2 | Обмен с использованием SQL-запросов | Обмен с использованием регламентированных запросов типа «SQL-запрос».  [Раздел 1.4.1](#_1.4.1_Обмен_с) содержит описание данного вида обмена |
| 3 | Обмен с использованием Рассылок | Обмен данными с использованием регламентированных запросов типа «Рассылка» в соответствии с созданной подпиской.  [Раздел 1.4.2](#_1.4.2_Обмен_с) содержит описание данного вида обмена |
| 4 | Обмен с использованием запросов к REST-сервису ИС Ответчика | Обмен с использованием регламентированных запросов типа «Rest-сервис» к зарегистрированным в ПОДД REST-сервисам ИС Ответчика.  [Раздел 1.4.3](#_1.4.3_Обмен_с) содержит описание данного вида обмена |

### 1.2.1 Классификация регламентированных SQL-запросов

Классификация Регламентированных SQL-запросов приведена в [Таблице 1.2](#_bb430a1bf96874db58083e69ffdf8fdf).

Таблица 1.2 Классификация Регламентированных SQL-запросов

|  |  |  |  |
| --- | --- | --- | --- |
| **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, содержащие 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 которых 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 регламентированных SQL-запросов 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».

## 1.3 Участники информационного обмена с использованием ПОДД СМЭВ

**Участник взаимодействия** — это орган или организация, участвующие в информационном обмене через СМЭВ.

[Рисунок - 1.1](#_8bd6c5af55b02c4feb3930eacab796ff) иллюстрирует общую схему взаимодействия участников.

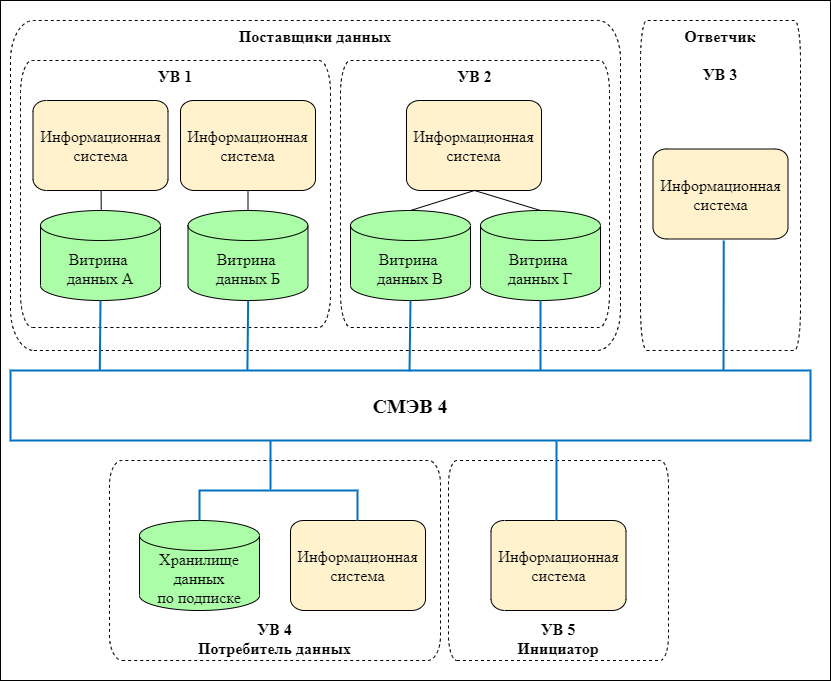


Рисунок - 1.1 Участники информационного обмена с использованием ПОДД СМЭВ

При участии в информационном обмене через ПОДД УВ может одновременно выступать как в роли Поставщика и/или Ответчика, так и в роли Потребителя данных и/или Инициатора запроса в зависимости от используемого вида обмена.

*В информационных обменах с использованием Витрин данных* УВ может иметь роль Поставщика и Потребителя. Каждый УВ может выступать одновременно как в качестве Поставщика данных, предоставляя данные со своих Витрин, так и в качестве Потребителя данных, выполняя запросы к другим УВ.

**Потребители данных** имеют возможность получать сведения из Витрины Поставщиков данных следующими способами:

1. используя регламентированный SQL-запрос (см. [Раздел 1.4.1](#_1.4.1_Обмен_с));
2. сформировав подписку на регламентированный запрос типа «Рассылка» и получая сведения в виде уведомлений об изменениях (см. [Раздел 1.4.2](#_1.4.2_Обмен_с)).

**Поставщики данных** обеспечивают доступ к актуальным данным на Витринах:

1. размещают и актуализируют данные на Витринах;
2. осуществляют подключение Витрин к ПОДД СМЭВ;

*В информационных обменах с использованием запросов к REST-сервису ИС Ответчика* (без использования Витрин) УВ может иметь роль Инициатора и Ответчика. Каждый УВ может выступать одновременно как в качестве Ответчика, предоставляя данные c сервиса ИС, так и в качестве Инициатора запроса, выполняя запросы к другим УВ.

**Инициаторы запроса** имеют возможность получать сведения из ИС Ответчика, выполнив запрос к REST-сервису ИС Ответчика (см. [Раздел 1.4.3](#_1.4.3_Обмен_с)).

**Ответчики** разворачивают REST-сервисы и предоставляют к нему доступ через ПОДД СМЭВ.

## 1.4 Информационный обмен участников взаимодействия с использованием ПОДД СМЭВ

Для осуществления информационного обмена с использованием ПОДД СМЭВ должно быть обеспечено подключение ИС УВ к ПОДД СМЭВ с помощью Агента СМЭВ4 соответствии с «Руководством администратора Агента СМЭВ4».

### 1.4.1 Обмен с использованием регламентированных SQL-запросов

#### 1.4.1.1 Общее описание информационного обмена

Регламентированные SQL-запросы формируются на основе Моделей данных Витрин и предназначены для получения Потребителем сведений из Витрины Поставщика данных.

Классификация регламентированных SQL-запросов приведена в [разделе 1.2.1](#_1.2.1_Классификация_регламентирован).

#### 1.4.1.2 Требования к участникам взаимодействия

##### 1.4.1.2.1 Требования к Поставщикам данных

Поставщик данных для участия в информационном обмене с использованием регламентированных SQL-запросов должен выполнить следующие требования.

Требования к поставщику данных для участия в информационном обмене с использованием регламентированных SQL-запросов приведены ниже в [Таблице 1.3](#_7845db6eb39a5a5dfc542a07a5a82396).

Таблица 1.3 Требования к Поставщикам данных

|  |  |  |
| --- | --- | --- |
| № | Требования | Комментарии [[2]](#footnote-2) |
| 1 | Зарегистрировать ИС в СМЭВ | В соответствии с документом «Правила и процедуры работы в подсистеме обеспечения доступа к данным федеральной государственной информационной системы «Единая система межведомственного электронного взаимодействия» по Методическим рекомендациям версии 4.XX» |
| 2 | Развернуть и настроить в своем контуре Агент СМЭВ4 | В соответствии с документом «Руководство администратора Агента СМЭВ4» |
| 3 | Обеспечить сетевую связанность Агента СМЭВ4, размещенного в инфраструктуре участника, и ПОДД СМЭВ |
| 4 | Развернуть в своем контуре ПО Витрина данных | Исходные коды ПО Витрина данных размещены в НФАП |
| 5 | Настроить ПО Витрина данных для взаимодействия с Агентом СМЭВ4 | См. [Раздел 2.2](#_2.2_Протокол_взаимодействия) |
| 6 | Загрузить модель данных ПО Витрина данных в ПОДД СМЭВ | Через ЕИП НСУД, в соответствии с документом «Инструкция по работе в ЕИП НСУД» |
| 7 | Связать ПО Витрина данных и ИС |
| 8 | Зарегистрировать Регламентированные SQL-запросы в ПОДД СМЭВ | Через ЕИП НСУД, в соответствии с документом «Инструкция по работе в ЕИП НСУД» |
| 9 | Добавить критерии доступа к Регламентированному SQL-запросу | Через ЛК УВ, в соответствии с документом «Руководство пользователя ЛК УВ» |

##### 1.4.1.2.2 Требования к Потребителям данных

Требования к потребителю данных для участия в информационном обмене с использованием регламентированных SQL-запросов приведены в [Таблице 1.4](#_32c21e29131e273962884a1a96ce0c81).

Таблица 1.4 Требования к Потребителям данных

|  |  |  |
| --- | --- | --- |
| № | Требования | Комментарии [[3]](#footnote-3) |
| 1 | Зарегистрировать ИС в СМЭВ | В соответствии с документом «Правила и процедуры работы в подсистеме обеспечения доступа к данным федеральной государственной информационной системы «Единая система межведомственного электронного взаимодействия» по Методическим рекомендациям версии 4.XX» |
| 2 | Развернуть и настроить в своем контуре Агент СМЭВ4 | В соответствии с документом «Руководство администратора Агента СМЭВ4» |
| 3 | Обеспечить сетевую связанность Агента СМЭВ4, размещенного в инфраструктуре участника, и ПОДД СМЭВ |
| 4 | Обеспечить свободный объем оперативной памяти на стороне Агента СМЭВ4 не менее размера передаваемых и получаемых данных |
| 5 | Настроить ИС, которая выполняет запросы и осуществляет обработку данных, получаемых от ПОДД СМЭВ, для взаимодействия с Агентом СМЭВ4 | См. [Раздел 2.3](#_2.3_Протокол_взаимодействия) |
| 6 | Получить полномочия на выполнение регламентированного SQL-запроса для своей ИС | В соответствии с документом «Руководство пользователя ЛК УВ» |

#### 1.4.1.3 Взаимодействие участников обмена

1. Взаимодействие Ядра ПОДД СМЭВ с Агентами СМЭВ4 осуществляется с использованием Протокола ПОДД СМЭВ.
2. Взаимодействие Агента Поставщика данных с Витриной Поставщика данных осуществляется с использованием зарезервированных топиков брокера сообщений Apache Kafka. [Раздел 2.2.1](#_2.2.1_Перечень_топиков) содержит перечень топиков Apache Kafka.
3. Взаимодействие ИС Потребителя с Агента Потребителя осуществляется через REST или JDBC-интерфейс в соответствии со спецификацией, описанной в [разделе 2.3](#_2.3_Протокол_взаимодействия).

[Рисунок - 1.2](#_c2af12ef4a09bf9a018ab59602502e0a) иллюстрирует общую схему взаимодействия участников обмена.

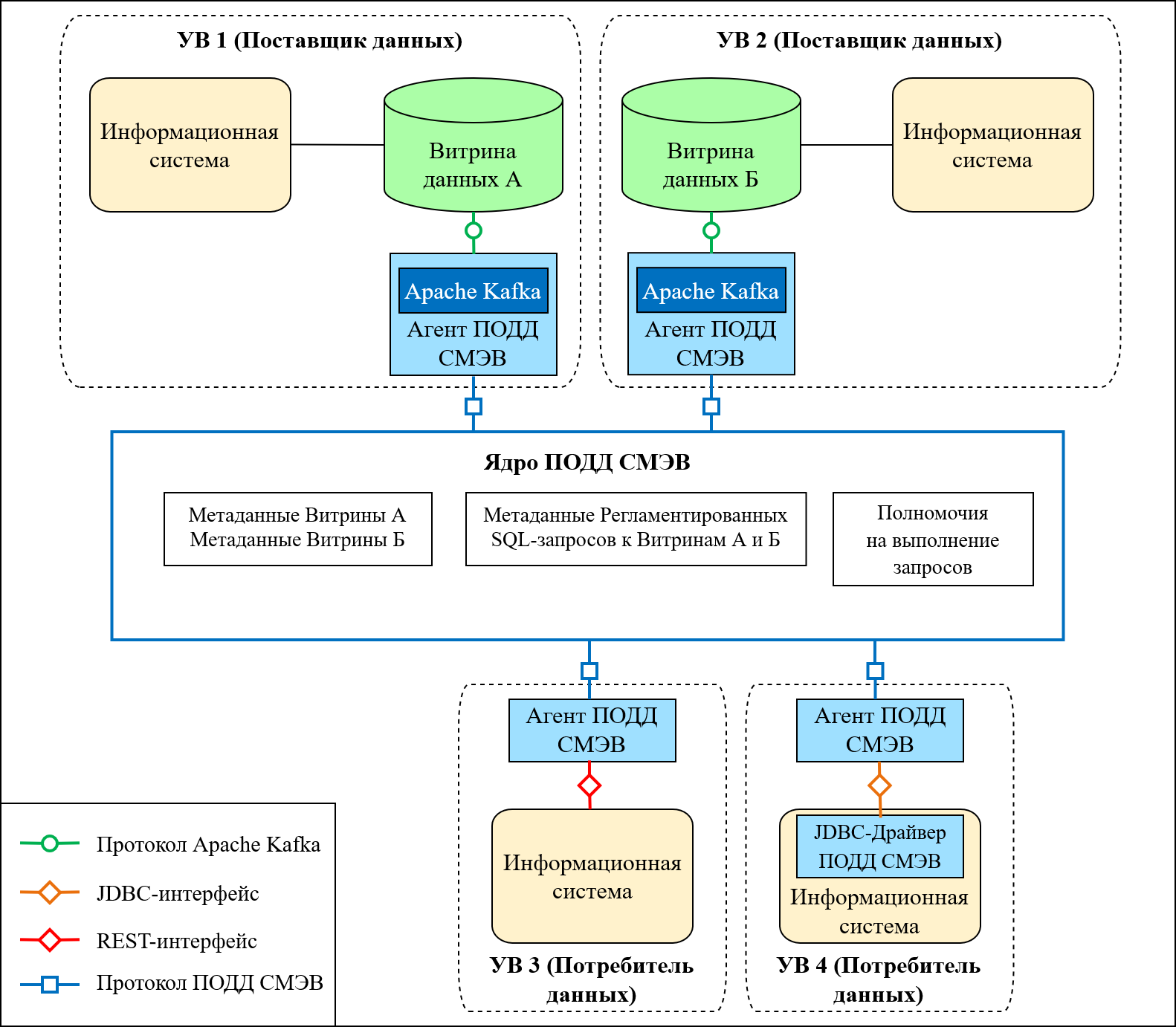


Рисунок - 1.2 Информационный обмен при выполнении запроса с использованием ПОДД СМЭВ

#### 1.4.1.4 Диаграмма последовательности информационного обмена

[Рисунок - 1.3](#_d5ed3ed9e149b918997a2d9c94d0c806) содержит диаграмму последовательности осуществления информационного обмена с использованием регламентированных SQL-запросов.

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

1. ИС Потребителя данных передаёт SQL-запрос Агенту СМЭВ4. Передача запроса может осуществляться с использованием:

* JDBC-подключения;
* REST-интерфейса.

Настройки подключения выполняются в рамках настройки ИС для взаимодействия с Агентом СМЭВ4 согласно «Руководству администратора Агента ПОДД СМЭВ»[[4]](#footnote-4).

[Раздел 2.3](#_2.3_Протокол_взаимодействия) содержит Спецификацию взаимодействия.

1. Агент Потребителя данных проверяет наличие включенного режима блокировки на отправку запросов. Дальнейшие шаги выполняются в случае отсутствия блокировки.
2. Агент Потребителя подписывает запрос ЭП ОВ.
3. Агент Потребителя передает полученный запрос в Ядро ПОДД СМЭВ.

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

1. Выполняет проверку ЭП ОВ, которой подписан запрос, проверку запроса на корректность и наличие у Потребителя доступа на выполнение запросов к данным. Если хотя бы одна из проверок возвращает отрицательный результат, то прекращает выполнение запроса и возвращает Агенту Потребителя соответствующую ошибку в качестве результата запроса.
2. Выполняет преобразование полученной мнемоники регламентированного SQL-запроса в SQL-выражение в соответствии с загруженным определением и формирует один или несколько запросов в адрес Витрин Поставщиков данных (далее – подзапросы).
3. 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 сообщение о включении режима блокировки. Агент Потребителя при получении такого сообщения активирует режим блокировки и отвечает ошибкой на запросы от Потребителя.

1. Передаёт каждый подзапрос в соответствующий Агент Поставщика.

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

1. Агент Поставщика данных получает запрос 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 доступа на выполнение Регламентированного SQL-запроса для Потребителя данных, отправившего запрос.

Если для запроса не подтвержден доступ, то Агент Поставщика прекращает выполнение запроса и возвращает Агенту Потребителя соответствующую ошибку в качестве результата запроса. При подтвержденном доступе 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. Витрина данных формирует результат выполнения подзапроса.
3. Агент Поставщика данных считывает результат выполнения подзапроса из зарезервированного топика брокера сообщений Apache Kafka.
4. Агент Поставщика подписывает результат ЭП ОВ.
5. Агент Поставщика формирует ответ и передает его в Ядро ПОДД СМЭВ.

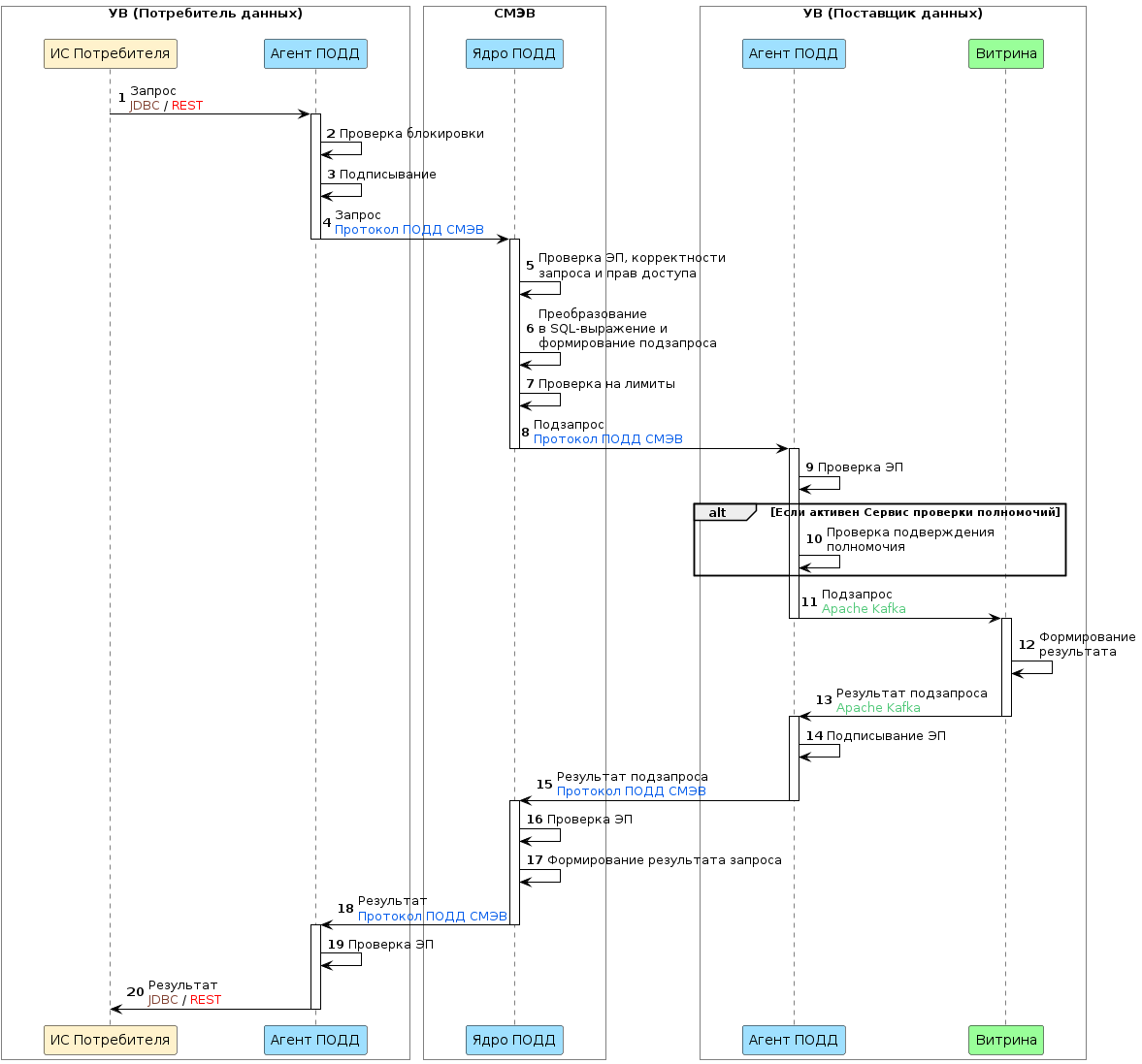


Рисунок - 1.3 Диаграмма последовательности процесса выполнения SQL-запроса

*Ядро ПОДД СМЭВ:*

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

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

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

### 1.4.2 Обмен с использованием Рассылок

#### 1.4.2.1 Общее описание информационного обмена

ПОДД СМЭВ и ПО «Витрина данных» позволяют организовать процесс автоматического:

* информирования Потребителя данных об изменениях на Витрине Поставщика данных посредством передачи уведомления об изменениях;
* размещения и актуализация данных из Витрины Поставщика данных в контуре Потребителя данных посредством снапшотов и дельт.

Такой обмен осуществляется по Подписке Потребителя данных на получение изменений данных с использованием регламентированного запроса типа «Рассылка»[[5]](#footnote-5).

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.

|  |
| --- |
| Примечание: |
| В настоящий момент подписка не может быть поставлена на паузу. Для остановки обмена данных по подписке она должна быть отменена. |

[Таблица 1.5](#_fa228b3fe9b909cae92209cc91dab772) и [Рисунок - 1.4](#_cd284620854f8b721f4e0b589781e406) демонстрируют виды подписок.

Таблица 1.5 Виды подписок

|  |  |  |
| --- | --- | --- |
|  | Вид | Характеристика |
| 1 | Простая | Подписка на один простой регламентированный запрос типа «Рассылка» |
| 2 | Множественная | Подписка на несколько простых регламентированных запросов типа «Рассылка» к таблицам одной Витрины данных |
| 3 | Распределённая | Подписка на один распределённый регламентированный запрос типа «Рассылка» |

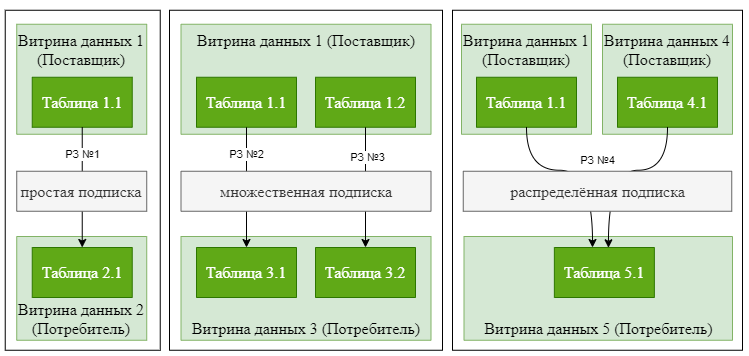


Рисунок - 1.4 Виды подписок

|  |
| --- |
| Примечание: |
| Множественная и распределённая подписки одновременно не работают на одной Витрине (Витрина 1 на [Рисунке - 1.4](#_cd284620854f8b721f4e0b589781e406) не может одновременно участвовать в подписках на схемах в центре и справа). |

Для использования регламентированного запроса типа «Рассылка» в подписке он должен удовлетворять следующим требованиям:

1. Общие требования:

1.1. Недопустимо использование в SQL-запросе параметров (за исключением статического where).

1.2. Недопустимо использование в SQL-запросе set operation («union», «minus», «intersect» и т.п.), оконных функций, коррелирующих подзапросов, группирующих функций, сортировок, common table expression, semi-join.

1.3. Используемые в запросе функции должны поддерживаться Витринами Поставщиков.

1. Дополнительные требования для нераспределённых подписок:

2.1. Количество источников данных в SQL-запросе должно быть равно одному.

1. Дополнительные требования для нераспределённых (множественных) подписок:

3.1. Количество источников данных в SQL-запросе должно быть равно одному.

3.2. Во всех SQL-запросах должна быть указана одна и та же Витрина (таблицы могут быть разными).

1. Дополнительные требования для распределённых подписок:

4.1. Количество источников данных должно быть равно двум (обработка большего количества в ПОДД не поддерживается).

4.2. Объединение данных в SQL-запросе должно осуществляться с помощью inner join (обработка outer join в ПОДД не поддерживается).

[Раздел 1.5.3](#_1.5.3_Подписки_Потребителей) содержит описание метаданных подписки.

Жизненный цикл подписки состоит из следующих этапов:

1. Подготовительные мероприятия для регистрации подписки в соответствии с [разделом 1.4.2.2](#_1.4.2.2_Требования_к);
2. Регистрация подписки в ПОДД и Витринах Поставщиков и Потребителей в соответствии с [разделом 1.4.2.4.1](#_1.4.2.4.1_Регистрация_подписки);
3. Осуществление информационного обмена по подписке в соответствии с [разделом 1.4.2.4.2](#_1.4.2.4.2_Осуществление_информацион), в том числе:

* уведомление о наличии дельты от Поставщика;
* передача уведомления о наличии дельты Потребителю;
* запрос у Поставщика дельты по инициативе Потребителя или ПОДД;
* передача дельты Потребителю;

1. Удаление подписки в ПОДД и Витринах Поставщиков в соответствии с [разделом 1.4.2.4.3](#_1.4.2.4.3_Удаление_подписки).

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

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

#### 1.4.2.2 Требования к участникам взаимодействия

##### 1.4.2.2.1 Требования к Поставщикам данных

Поставщик данных для участия в информационном обмене по Рассылке должен выполнить следующие требования.

Таблица 1.6 Требования к Поставщикам данных для использования Рассылок

|  |  |  |
| --- | --- | --- |
| № | Требование | Инструкция [[6]](#footnote-6) |
| 1 | Зарегистрировать ИС в СМЭВ | Инструкция приведена в документе «Правила и процедуры работы в подсистеме обеспечения доступа к данным федеральной государственной информационной системы «Единая система межведомственного электронного взаимодействия» по Методическим рекомендациям версии 4.XX» |
| 2 | Развернуть и настроить в своем контуре Агент СМЭВ4 | Инструкция приведена в документе «Руководство администратора Агента ПОДД СМЭВ» |
| 3 | Обеспечить сетевую связанность Агента СМЭВ4, размещенного в инфраструктуре участника, и ПОДД СМЭВ | Инструкция приведена в документе «Руководство администратора Агента ПОДД СМЭВ» |
| 4 | Развернуть в своем контуре ПО Витрина данных | Исходные коды ПО Витрина данных размещены в НФАП |
| 5 | Настроить Витрины для взаимодействия с Агентом СМЭВ4 | См. [Раздел 2.2](#_2.2_Протокол_взаимодействия) |
| 6 | Связать ПО Витрина данных и ИС | Через ЕИП НСУД, в соответствии с документом «Инструкция по работе в ЕИП НСУД» |
| 7 | Загрузить модель данных Витрины в ПОДД СМЭВ | Инструкция приведена в документе «Правила и процедуры работы в подсистеме обеспечения доступа к данным федеральной государственной информационной системы «Единая система межведомственного электронного взаимодействия» по Методическим рекомендациям версии 4.XX» |
| 8 | Зарегистрировать Регламентированный запрос типа «Рассылка», по которому будет выполняться подписка в ПОДД СМЭВ | Через ЛК УВ, в соответствии с документом «Руководство пользователя ЛК УВ» |
| 9 | Добавить критерии доступа к Регламентированному запросу типа «Рассылка» |
| 10 | Согласовать право доступа Потребителя к Регламентированному запросу типа «Рассылка», по которому будет выполняться Подписка | Инструкция приведена в документе «Правила и процедуры работы в подсистеме обеспечения доступа к данным федеральной государственной информационной системы «Единая система межведомственного электронного взаимодействия» по Методическим рекомендациям версии 4.XX» |
| 11 | Зарегистрировать Подписку в ПОДД СМЭВ | Инструкция приведена в документе «Правила и процедуры работы в подсистеме обеспечения доступа к данным федеральной государственной информационной системы «Единая система межведомственного электронного взаимодействия» по Методическим рекомендациям версии 4.XX» |

##### 1.4.2.2.2 Требования к Потребителям данных

Потребитель данных для участия в информационном обмене по Рассылке должен выполнить следующие требования.

Таблица 1.7 Требования к Потребителям данных для использования Рассылок

| № | Требование | Инструкция [[7]](#footnote-7) |
| --- | --- | --- |
| 1 | Зарегистрировать ИС в СМЭВ | Инструкция приведена в документе «Правила и процедуры работы в подсистеме обеспечения доступа к данным федеральной государственной информационной системы «Единая система межведомственного электронного взаимодействия» по Методическим рекомендациям версии 4.XX» |
| 2 | Разместить в своем контуре Агент СМЭВ4, выполнить настройки для соответствующего вида обменов | Инструкция приведена в документе «Руководство администратора Агента ПОДД СМЭВ» |
| 3 | Обеспечить сетевую связанность Агента СМЭВ4, размещенного в инфраструктуре участника, и ПОДД СМЭВ |
| 4 | Обеспечить свободный объем оперативной памяти на стороне Агента СМЭВ4 не менее размера передаваемых и получаемых данных |
| 5 | Развернуть в своем контуре ПО Витрина данных в качестве хранилища данных по подписке | Исходные коды Витрины размещены в НФАП |
| 6 | Настроить Витрину для взаимодействия с Агентом СМЭВ4 | См [Раздел 2.4](#_2.4_Протокол_взаимодействия) |
| 7 | Согласовать право доступа к Регламентированному запросу типа «Рассылка», по которому будет выполняться подписка с Поставщиком данных | Инструкция приведена в документе «Правила и процедуры работы в подсистеме обеспечения доступа к данным федеральной государственной информационной системы «Единая система межведомственного электронного взаимодействия» по Методическим рекомендациям версии 4.XX» |

#### 1.4.2.3 Взаимодействия участников обмена

1. Взаимодействие Ядра ПОДД СМЭВ с Агентами СМЭВ4 осуществляется с использованием Протокола ПОДД СМЭВ.
2. Взаимодействие Агента Поставщика данных с Витриной Поставщика данных осуществляется с использованием зарезервированных топиков брокера сообщений Apache Kafka. [Раздел 2.2.1](#_2.2.1_Перечень_топиков) содержит перечень топиков Apache Kafka.
3. Взаимодействие Агента Потребителя данных с Витриной по подписке осуществляется с использованием зарезервированных топиков брокера сообщений Apache Kafka. [Раздел 2.4.1](#_2.4.1_Перечень_топиков) содержит перечень топиков Apache Kafka;

[Рисунок - 1.5](#_94df3f676923bb5f33385f1133c140aa) иллюстрирует общую схему взаимодействия участников обмена.

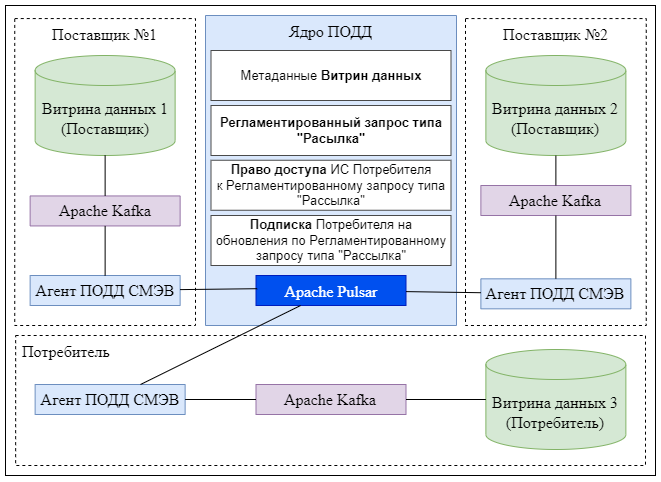


Рисунок - 1.5 Информационный обмен с использованием Рассылок

#### 1.4.2.4 Описание этапов

##### 1.4.2.4.1 Регистрация подписки

[Рисунок - 1.6](#_d3868152c1dfc08dfca0e29dda3102b0) демонстрирует порядок регистрации подписки, соответствующее описание процесса приведено ниже.

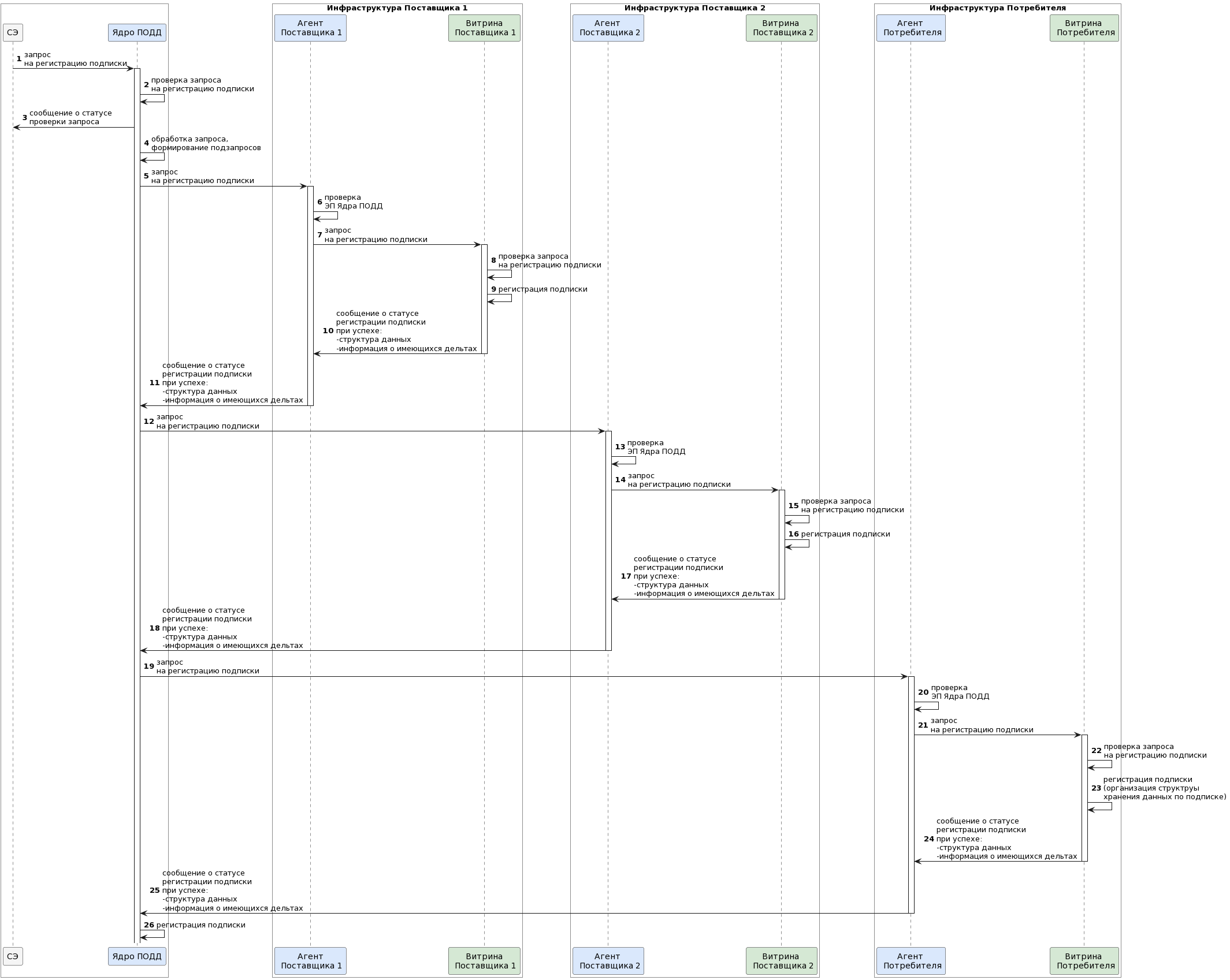


Рисунок - 1.6 Диаграмма последовательности – Регистрация подписки

1. Запрос на регистрацию подписки передаётся в ПОДД в соответствии с [разделом 1.5.3](#_1.5.3_Подписки_Потребителей).

*Ядро ПОДД СМЭВ:*

1. Осуществляет следующие проверки:

* проверка уникальности идентификатора подписки в запросе;
* в случае множественной подписки – проверка регламентированных запросов типа «Рассылка» на то, что они нераспределённые и источником во всех запросах является одна Витрина;
* проверка регламентированного запроса типа «Рассылка» на его наличие в ПОДД;
* проверка соответствия регламентированного запроса типа «Рассылка» требованиям для использования в подписке (приведены в [разделе 1.4.2.1](#_1.4.2.1_Общее_описание));
* проверка соответствия регламентированных запросов на использование допустимого для информационного обмена по подписке количество Поставщиков;
* проверка наличия зарегистрированных в ПОДД Витрин Поставщиков и Потребителя;
* проверка наличия прав доступа ИС Потребителя к регламентированному запросу типа «Рассылка», указанному в регистрируемой подписке;
* проверка, является ли действующей версия регламентированного запроса типа «Рассылка»;
* в случае распределённой подписки – проверка на отсутствие значения в поле «cronExpression».

1. Отправляет ответ со статусом обработки запроса.
2. Определяет перечень подзапросов:

* если подписка нераспределённая, то формируется один подзапрос на получение дельты (тип - DATA);
* если подписка распределённая, формирует набор подзапросов:
  + подзапрос ключей Витрины Поставщика 1 (тип - KEYS);
  + подзапрос ключей Витрины Поставщика 2 (тип - KEYS);
  + подзапрос ключей Витрины Поставщика 1 с фильтрацией по ключам Витрины Поставщика 2 (тип - KEYS\_FILTERED);
  + подзапрос ключей Витрины Поставщика 2 с фильтрацией по ключам Витрины Поставщика 1 (тип - KEYS\_FILTERED);
  + подзапрос дельты Витрины Поставщика 1 с фильтрацией по ключам Витрины Поставщика 2 (тип - DATA\_FILTERED);
  + подзапрос дельты Витрины Поставщика 2 с фильтрацией по ключам Витрины Поставщика 1 (тип - DATA\_FILTERED);
  + подзапрос объединения частей дельт на Потребителе (тип - FINAL).

1. (12) Отправляет запрос на регистрацию подписки всем участвующим в подписке Поставщикам в соответствующие Агенты СМЭВ4.

*Агент Поставщика данных:*

1. (13) Осуществляет проверку ЭП Ядра ПОДД СМЭВ;
2. (14) передает запрос на регистрацию подписки Витрине в топик <префикс>.replication.rq.

*Витрина Поставщика данных:*

1. (15) Осуществляет проверку запроса на регистрацию подписки:

* проверка уникальности идентификатора подписки;
* проверка наличия витрины, указанной в подписке;
* проверка наличия таблицы, указанной в подписке;
* проверка корректности sql-запроса;
* проверка соответствия полей sql-запроса таблице Витрины Поставщика;
* проверка наличия атрибутов «primaryKey» и «shardingKey» в структуре итоговой таблицы.

1. (16) регистрирует подписку (устанавливает статус подписки – активная (OPENED))
2. (17) отправляет ответ в Агент Поставщика данных:

* в случае успешной обработки запроса – структуру таблиц в топик <префикс>.replication.rs.
* в случае неуспешной обработки запроса – уведомление об ошибке в топик <префикс>.replication.err.

*Агент Поставщика данных:*

1. Пересылает структуру таблиц в Ядро ПОДД СМЭВ.

*Ядро ПОДД СМЭВ:*

1. Пересылает структуру таблиц Агенту Потребителя данных.

*Агент Потребителя данных:*

1. Осуществляет проверку ЭП Ядра ПОДД СМЭВ;
2. Передает в Хранилище данных по подписке структуру таблиц с использованием топика <префикс>.replication.in.rq.

*Витрина Потребителя данных:*

1. Осуществляет проверку запроса на регистрацию подписки:

* проверка уникальности идентификатора подписки;
* проверка уникальности целевой таблицы подписки;
* проверка корректности sql-запроса на объединение данных.

1. Регистрирует подписку (устанавливает статус подписки – активная (OPENED))
2. Отправляет ответ в Агент Потребителя данных:

* в случае успешной обработки - уведомление об успешном создании структуры данных в топик <префикс>.replication.in.rs.
* в случае неуспешной обработки - уведомление об ошибке в топик <префикс>.replication.in.err.

*Агент Потребителя данных:*

1. Отправляет в Ядро ПОДД СМЭВ статус обработки структуры таблиц Хранилищем данных.

*Ядро ПОДД СМЭВ:*

1. Регистрирует подписку.

##### 1.4.2.4.2 Осуществление информационного обмена по Рассылке

[Рисунок - 1.7](#_49020291285287eba101cc4a60c8abae) иллюстрирует порядок информационного обмена по рассылке, соответствующее описание процесса приведено ниже. Загрузка снапшота осуществляется аналогично последующей загрузке дельт.

*Витрина Поставщика данных:*

1. Передаёт в Агент Поставщика уведомление о наличии дельты в топик <префикс>.delta.notification.

*Агент Поставщика данных:*

1. Передаёт уведомление в Ядро ПОДД СМЭВ.

*Ядро ПОДД СМЭВ:*

1. По наличию расписания (cronExpression) определяет, кто является инициатором запроса дельты – Ядро ПОДД или Потребитель[[8]](#footnote-8). В случае распределённых подписок расписание не задаётся, инициатором запроса дельты всегда является Потребитель.

* если расписание задано, осуществляет запрос дельты самостоятельно (переход к шагу 11). При этом если Потребитель еще не прислал ответ о результате применения предыдущего пакета дельт, новый не запрашивается.
* если расписание не задано, осуществляет отправку уведомления Потребителю (переход к шагу 4).

1. Отправляет уведомление о наличии дельты в Агент Потребителя и не запрашивает дельту, пока не придёт команда от Потребителя.

*Агент Потребителя данных:*

1. Передаёт уведомление о наличии дельты Витрине Потребителя в топик <префикс>.delta.notification.in.

*Витрина Потребителя данных:*

1. Обрабатывает уведомление о наличии новой дельты;
2. В случае распределённой подписки определяет порядок обращения к Поставщикам;
3. Посылает запрос на получение дельты в командный топик <префикс>.command.podd.

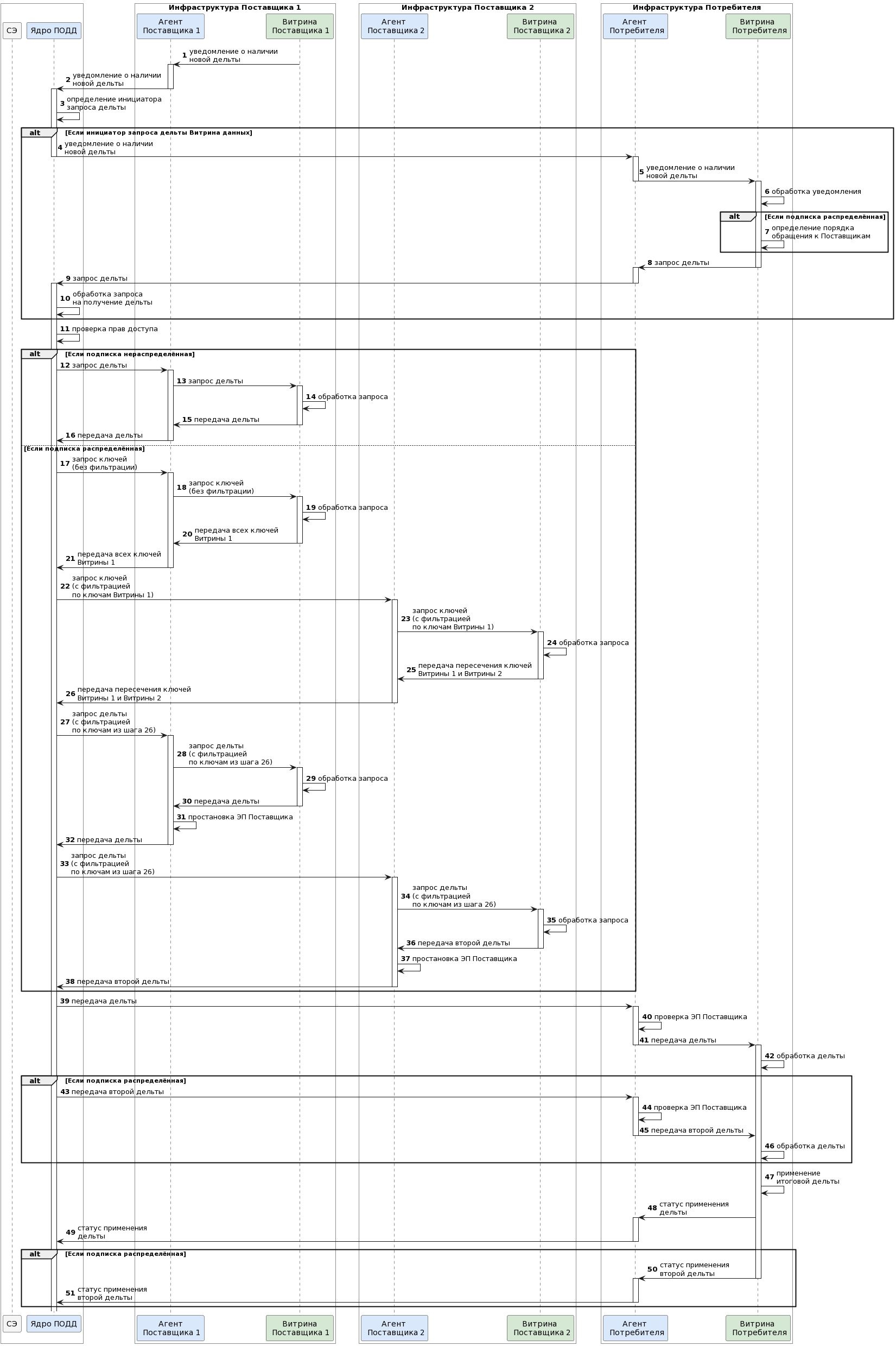


Рисунок - 1.7 Диаграмма последовательности – Информационный обмен по рассылке

*Агент Потребителя данных:*

1. Передаёт запрос в Ядро ПОДД СМЭВ.

*Ядро ПОДД СМЭВ:*

1. Обрабатывает запрос на получение дельты;
2. Проверяет права доступа Потребителя данных.

**Если подписка нераспределённая:**

*Ядро ПОДД СМЭВ:*

1. Передаёт запрос дельты на Агент Поставщика.

*Агент Поставщика данных:*

1. Передаёт запрос дельты Витрине Поставщика в топик <префикс>.delta.rq.

*Витрина данных Поставщика:*

1. Обрабатывает запрос дельты;
2. Передаёт дельту в Агент Поставщика:

* в случае успешной обработки запроса результат в топик <префикс>.delta.rs.
* в случае неуспешной обработки запроса - уведомление об ошибке в топик <префикс>.delta.err.

*Агент Поставщика данных:*

1. Передаёт дельту в Ядро ПОДД СМЭВ.

**Если подписка распределённая:**

*Ядро ПОДД СМЭВ:*

1. Передаёт запрос ключей (без фильтрации) в Агент Поставщика 1

*Агент Поставщика данных 1:*

1. Передаёт запрос ключей (без фильтрации) Витрине Поставщика в топик <префикс>.delta.rq.

*Витрина Поставщика данных 1:*

1. Обрабатывает запрос;
2. Передаёт дельту в Агент Поставщика 1:

* в случае успешной обработки запроса результат в топик <префикс>.delta.rs.
* в случае неуспешной обработки запроса - уведомление об ошибке в топик <префикс>.delta.err.

*Агент Поставщика данных 1:*

1. Передаёт ключи в Ядро ПОДД СМЭВ.

*Ядро ПОДД СМЭВ:*

1. Передаёт запрос ключей (с фильтрацией по ключам Витрины 1) в Агент Поставщика 2.

*Агент Поставщика данных 2:*

1. Передаёт запрос ключей (с фильтрацией по ключам Витрины 1) Витрине Поставщика 2 в топик <префикс>.delta.tp.

*Витрина Поставщика данных 2:*

1. Обрабатывает запрос;
2. Передаёт пересечение ключей в Агент Поставщика 2:

* в случае успешной обработки запроса результат в топик <префикс>.delta.rs.
* в случае неуспешной обработки запроса - уведомление об ошибке в топик <префикс>.delta.err.

*Агент Поставщика данных 2:*

1. Передаёт пересечение ключей в Ядро ПОДД СМЭВ.

*Ядро ПОДД СМЭВ:*

1. (33) Передаёт запрос дельты (с фильтрацией по ключам) в Агент Поставщика 1(2).

*Агент Поставщика данных 1 (2):*

1. (34) Передаёт запрос дельты (с фильтрацией по ключам) в Витрине Поставщика 1(2).

*Витрина Поставщика данных 1 (2):*

1. (35) Обрабатывает запрос;
2. (36) Передаёт дельту в Агент Поставщика 1 (2).

*Агент Поставщика данных 1 (2):*

1. (37) Осуществляет приостановку ЭП Поставщика 1 (2)
2. (38) Передаёт пересечение ключей в Ядро ПОДД СМЭВ.

*Ядро ПОДД СМЭВ:*

1. Передаёт дельту в Агент Потребителя

*Агент Потребителя данных*

1. (44) Проверяет ЭП Поставщика;
2. (45) Передаёт дельту в Витрину Потребителя.

*Витрина Потребителя данных:*

1. (46) Обрабатывает полученную дельту;
2. Применяет дельту;
3. (50) Передаёт статус применения дельты.

*Агент Потребителя данных*

1. (51) Передаёт статус применения дельты в Ядро ПОДД СМЭВ.

##### 1.4.2.4.3 Удаление подписки

[Рисунок - 1.8](#_aaea3c372aff54417e558e0e76089ed3) иллюстрирует порядок отмены подписки, соответствующее описание процесса приведено ниже.

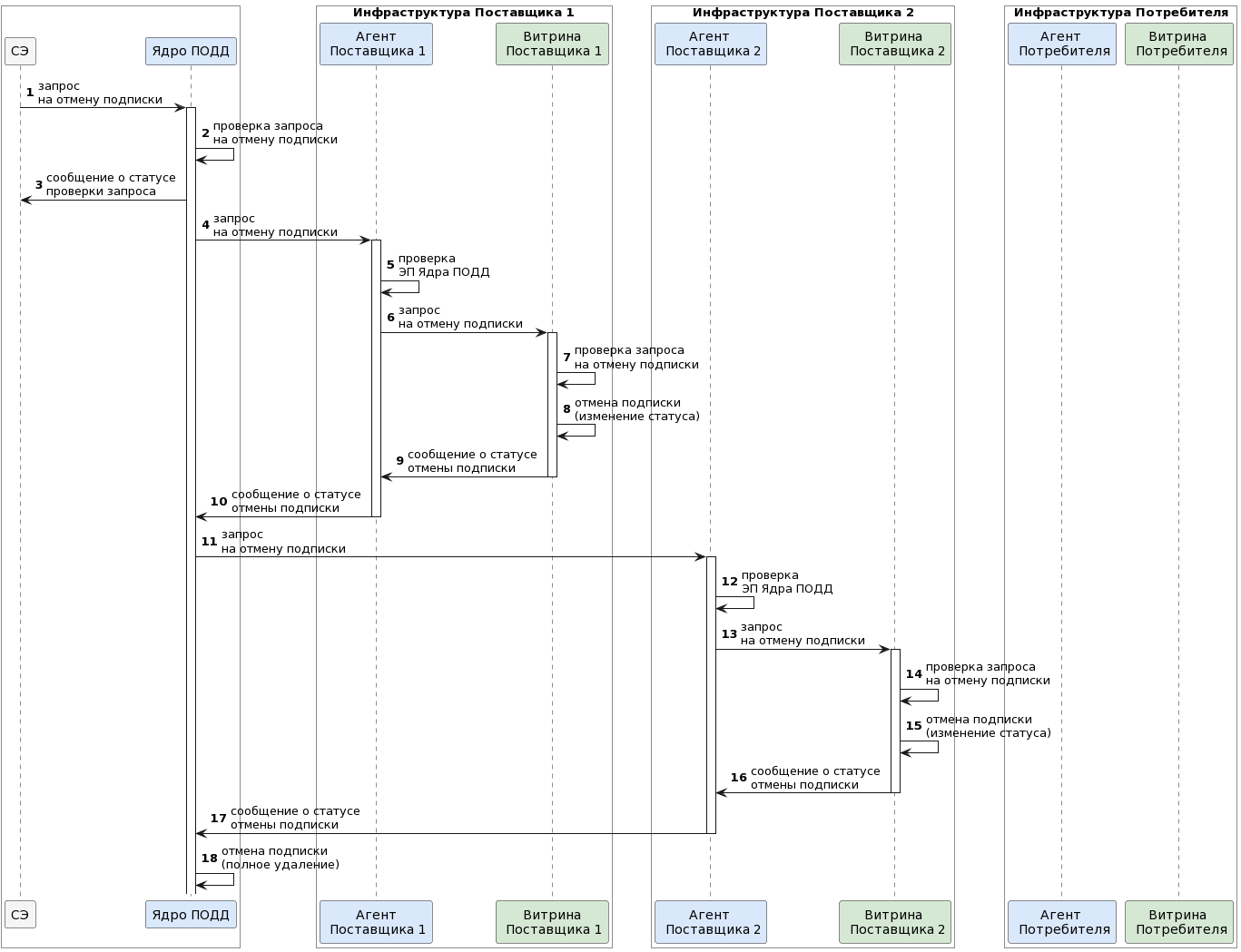


Рисунок - 1.8 Диаграмма последовательности – Удаление подписки

1. Запрос на удаление подписки передаётся в ПОДД СМЭВ в соответствии c [разделом 1.5.3](#_1.5.3_Подписки_Потребителей).

*Ядро ПОДД СМЭВ:*

1. Осуществляет проверку запрос на отмену подписки на наличие в ПОДД СМЭВ подписки с указанным идентификатором;
2. Отправляет ответ со статусом обработки запроса;
3. (11) Отправляет запрос на отмену подписки всем участвующим в подписке Поставщикам в соответствующие Агенты.

*Агент Поставщика данных:*

1. (12) Осуществляет проверку ЭП Ядра ПОДД СМЭВ;
2. (13) Отправляет запрос Витрине Поставщика в топик <префикс>.replication.cancel.rq.

*Витрина Поставщика данных:*

1. (14) Осуществляет проверку запроса на отмену подписки;
2. (15) Отменяет подписку (устанавливает статус подписки – отменённая (CANCELED));
3. (16) Отправляет ответ в Агент Поставщика данных в топик <префикс>.replication.cancel.rs.

*Агент Поставщика данных:*

1. (17) Отправляет в Ядро ПОДД СМЭВ статус обработки запроса на отмену подписки.

*Ядро ПОДД СМЭВ:*

1. Удаляет подписку после получения ответа от всех Поставщиков вне зависимости от статуса этих ответов (успех/ошибка). При этом статус Поставщика временно (до удаления) в случае успеха меняется на CANCELLED, а в случае ошибки – ERROR.

### 1.4.3 Обмен с использованием запросов к REST-сервису ИС Ответчика

#### 1.4.3.1 Общее описание информационного обмена

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

Также данный механизм обеспечивает:

* регистрацию REST-сервисов ИС Ответчиков, описываемых спецификацией OpenAPI, в ПОДД СМЭВ через ЛК УВ;
* разграничение доступа Инициаторов к REST-сервисам ИС Ответчиков путём использования принципа выдачи/изъятия прав на взаимодействие ИС Потребителя с REST-сервисом ИС Ответчика через ЛК УВ;
* форматно-логический контроль запросов ИС Инициаторов и ответов ИС Ответчиков на соответствие спецификации OpenAPI зарегистрированного в ПОДД СМЭВ REST-сервиса.
* механизм лимитирования запросов к REST-сервисам ИС Ответчика.

#### 1.4.3.2 Требования к участникам взаимодействия

##### 1.4.3.2.1 Требования к Ответчикам

Ответчик для участия в информационном обмене с использованием запросов к REST-сервису ИС Ответчика должен выполнить следующие требования.

Таблица 1.8 Требования к Ответчикам

| № | Требование | Комментарий [[9]](#footnote-9) |
| --- | --- | --- |
| 1 | Зарегистрировать ИС в СМЭВ | В соответствии с документом «Правила и процедуры работы в подсистеме обеспечения доступа к данным федеральной государственной информационной системы «Единая система межведомственного электронного взаимодействия» по Методическим рекомендациям версии 4.XX» |
| 2 | Развернуть и настроить в своем контуре Агент СМЭВ4 | В соответствии с документом «Руководство администратора Агента ПОДД СМЭВ» |
| 3 | Обеспечить сетевую связанность Агента СМЭВ4, размещенного в инфраструктуре участника, и ПОДД СМЭВ |
| 4 | Развернуть в своем контуре REST-сервис ИС Ответчика |  |
| 5 | Настроить REST-сервис ИС Ответчика, который обрабатывает получаемые от ПОДД СМЭВ запросы, для взаимодействия с Агентом СМЭВ4 | См. [Раздел 2.5](#_2.5_Протокол_взаимодействия) |
| 6 | Зарегистрировать REST-сервис ИС Ответчика (Регламентированный запрос типа «REST-сервис») в ПОДД СМЭВ | Через ЛК УВ, в соответствии с документом «Руководство пользователя ЛК УВ» |
| 7 | Добавить критерии доступа для запросов к развернутому REST-сервису |
| 8 | Для обеспечения возможности получения большого запроса размером до 30 Гб необходимо обновить версию Агента СМЭВ4 до версии не ниже 3.8.0. | В соответствии с документом «Руководство администратора Агента ПОДД СМЭВ» |
| 9 | При необходимости задать лимиты на выполнение запросов через Службу эксплуатации. Лимиты задаются в разрезе конкретного REST-сервиса ИС Ответчика и ИС Инициатора. Если лимиты не заданы, запросы ИС Инициатора к REST-сервису ИС Ответчика не ограничиваются.  Для использования механизма лимитирования необходимо:   * установить версию Агента СМЭВ4 на стороне Ответчика не ниже 3.8.0; * установить версию Агента СМЭВ4 на стороне Инициатора не ниже 3.8.0; * ИС Инициатора должна осуществлять запросы к Агенту Инициатора через порт для версии 3.8.0 или новее (см. «Руководство администратора Агента ПОДД СМЭВ»). | В соответствии с документом «Руководство администратора Агента ПОДД СМЭВ» |

##### 1.4.3.2.2 Требования к Инициаторам запросов

Инициатор запроса для участия в информационном обмене с использованием запросов к REST-сервису ИС Ответчика должен выполнить следующие требования.

Таблица 1.9 Требования к Инициаторам запросов

| № | Требование | Комментарий [[10]](#footnote-10) |
| --- | --- | --- |
| 1 | Зарегистрировать ИС в СМЭВ | В соответствии с документом «Правила и процедуры работы в подсистеме обеспечения доступа к данным федеральной государственной информационной системы «Единая система межведомственного электронного взаимодействия» по Методическим рекомендациям версии 4.XX» |
| 2 | Развернуть и настроить в своем контуре Агент СМЭВ4 | В соответствии с документом «Руководство администратора Агента ПОДД СМЭВ» |
| 3 | Обеспечить сетевую связанность Агента СМЭВ4, размещенного в инфраструктуре участника, и ПОДД СМЭВ |
| 4 | Настроить ИС, которая выполняет запросы и осуществляет обработку данных, получаемых от ПОДД СМЭВ, для взаимодействия с Агентом СМЭВ4 | См. [Раздел 2.3.2](#_2.3.2_REST-интерфейс_Агента) |
| 5 | Получить полномочия на выполнение запросов к REST-сервису Ответчика (Регламентированному запросу типа «REST-сервис») | Через ЛК УВ, в соответствии с документом «Руководство пользователя ЛК УВ» |
| 6 | Для обеспечения возможности отправки от инициатора большого запроса размером до 30 Гб необходимо:   * установить версию Агента СМЭВ4 на стороне Ответчика не ниже 3.8.0; * установить версию Агента СМЭВ4 на стороне Инициатора не ниже 3.8.0; * ИС Инициатора должна осуществлять запросы к Агенту Инициатора через порт для версии 3.8.0 или новее (см. «Руководство администратора Агента ПОДД СМЭВ»); * обеспечить свободный объем оперативной памяти на стороне Агента Инициатора не менее размера передаваемых данных.   Для версий Агента СМЭВ4 ранее 3.8.0 максимальный размер запроса от инициатора при обращении к REST-сервису ИС Ответчика ограничен 5 Мб. | В соответствии с документом «Руководством администратора Агента ПОДД СМЭВ» |

#### 1.4.3.3 Взаимодействие участников обмена

Механизм обмена с использованием запросов к REST-сервису ИС Ответчика реализован в виде комплекса взаимодействующих программных средств и артефактов, приведенных на [Рисунке - 1.9](#_ba5a43aed062f94269fef5cdc250e168).

1. Взаимодействие Ядра ПОДД с Агентами СМЭВ4 осуществляется по Протоколу ПОДД СМЭВ.
2. Взаимодействие Агента Инициатора с ИС Инициатора осуществляется с использованием REST-интерфейса в соответствии с [разделе 2.3.2](#_2.3.2_REST-интерфейс_Агента).
3. Взаимодействие Агента Ответчика с ИС Ответчика осуществляется в соответствии со спецификацией OpenAPI, загруженной в ПОДД СМЭВ (см. [Раздел 2.5](#_2.5_Протокол_взаимодействия)).

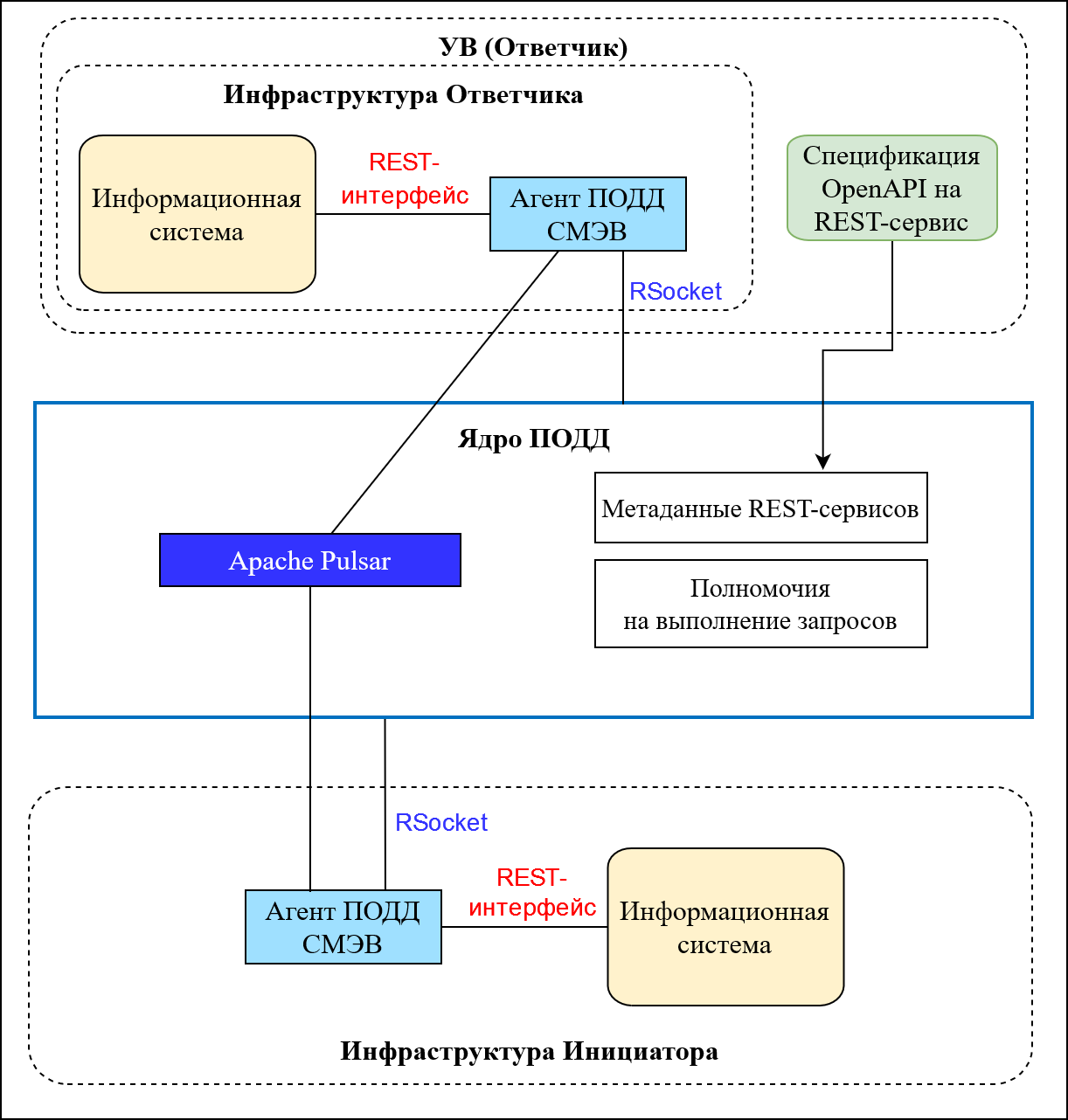


Рисунок - 1.9 Информационный обмен с использованием запросов к REST-сервису ИС Ответчика через ПОДД СМЭВ

#### 1.4.3.4 Описание этапов

##### 1.4.3.4.1 Регистрация в ПОДД REST-сервиса ИС Ответчика

Подготовка УВ к использованию механизма запросов к REST-сервисам ИС Ответчика описана в [разделе 1.4.3.2](#_1.4.3.2_Требования_к).

Этапы и основные шаги процесса:

*Сотрудник УВ*:

1. Реализует REST-сервис.
2. Осуществляет подготовку спецификации OpenAPI в соответствии с реализованным REST-сервисом.

|  |
| --- |
| Примечание: |
| основные положения по подготовке спецификаций OpenAPI для использования в ПОДД СМЭВ приведены в [разделе 1.5.5](#_1.5.5_REST-сервисы_ИС). |

1. Регистрирует REST-сервис своей ИС в ПОДД СМЭВ через ЛК УВ.

*Ядро ПОДД СМЭВ:*

1. Сохраняет метаданные REST-сервиса ИС Ответчика и отправляет подтверждение регистрации.

##### 1.4.3.4.2 Информационный обмен для механизма без возможности отправки большого запроса

*Взаимодействие ИС Инициатора и ИС Ответчика при обмене с использованием запросов к REST-сервису ИС Ответчика* ([Рисунок - 1.10](#_1b9d79f0d536474e8992f49216c35825)) *без возможности отправки большого запроса:*

*Агент ПОДД Инициатора запроса:*

1. Получает запрос к требуемому REST-сервису ИС Ответчика, сформированный в ИС Инициатора, с использованием REST-интерфейса. Спецификация взаимодействия приведена в [разделе 2.2.3](#_2.2.3_Примеры_реализации).
2. Отправляет метаданные запроса в Ядро ПОДД СМЭВ.

*Ядро ПОДД СМЭВ:*

1. Проверяет, что запрос и его параметры соответствуют спецификации OpenAPI REST-сервиса ИС Ответчика.
2. Отправляет результат проверки Агенту Инициатора.

*Агент Инициатора:*

1. Подписывает сообщение ЭП Инициатора.
2. Передает запрос в Ядро ПОДД СМЭВ.

*Ядро ПОДД СМЭВ:*

1. Выполняет проверку ЭП Инициатора.
2. Выполняет проверку наличия у Инициатора, 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 Инициатора соответствующую ошибку в качестве результата запроса.
3. Формирует и отправляет запрос Агенту Ответчика.

*Агент Ответчика:*

1. Проверяет ЭП Инициатора, которой подписан запрос.
2. Извлекает из сообщения запрос и формирует абсолютный URL-адрес запроса к соответствующему методу REST-сервиса ИС Ответчика.
3. Отправляет запрос REST-сервису ИС Ответчика согласно полученному URL адресу.

*REST-сервис ИС Ответчика:*

1. Выполняет полученный запрос (поиск и извлечение требуемых данных, создание или модификация сущностей БД и другое).
2. Возвращает в Агент Ответчика 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.

*Агент Ответчика:*

1. Формирует ответ и подписывает его ЭП Ответчика.
2. Передает ответ в Ядро ПОДД СМЭВ.

*Ядро ПОДД СМЭВ:*

1. Выполняет проверку ЭП Ответчика.
2. Перенаправляет результат Агенту Инициатора.

*Агент Инициатора:*

1. Выполняет проверку ЭП Ответчика.
2. Извлекает из сообщения HTTP код и тело (если присутствует) и возвращает их в качестве ответа ИС Потребителя с использованием REST-интерфейса.

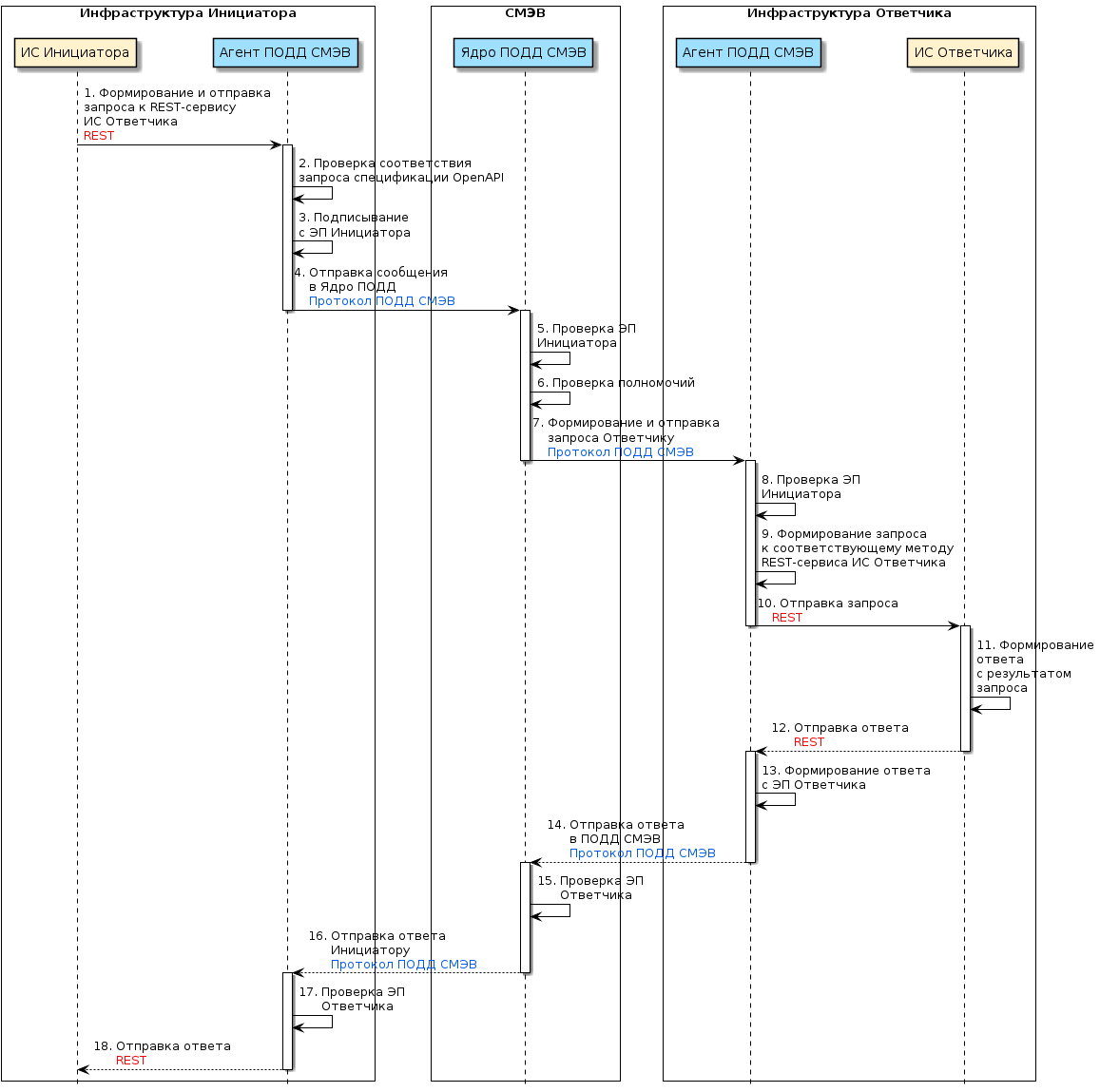


Рисунок - 1.10 Диаграмма информационного обмена с использованием запросов к REST-сервису ИС Ответчика через ПОДД СМЭВ

##### 1.4.3.4.3 Информационный обмен для механизма с возможностью отправки большого запроса

*Взаимодействие ИС Инициатора и ИС Ответчика при выполнении запроса к REST-сервису ИС Ответчика* ([Рисунок - 1.11](#_4aa7751e1784725a877ea3f8ac677aed)):

*Агент Инициатора:*

1. Получает запрос к REST-сервису ИС Ответчика.
2. Проверяет наличие активной блокировки на отправку запросов.
3. В случае отсутствия блокировки делит запрос на чанк метаданных и чанки тела запроса.
4. Подписывает чанк метаданных ЭП Инициатора.
5. Передает чанк метаданных в Ядро ПОДД СМЭВ.

*Ядро ПОДД СМЭВ:*

1. Осуществляет проверку ЭП Инициатора для чанка метаданных.
2. Осуществляет поиск спецификации OpenAPI REST-сервиса ИС Ответчика и валидацию на соответствие спецификации по чанку метаданных запроса:

* валидация осуществляется по параметрам, заголовкам и path;
* тело запроса не валидируется.

1. Проверяет наличие у Инициатора прав доступа для обращения к REST-сервису Ответчика.
2. Выполняет проверку соответствия ограничениям, заданным на REST-сервис ИС Ответчика:

* на количество запросов от ИС Инициатора за интервал времени;
* на объем данных, передаваемых ИС Потребителя по всем запросам за интервал времени.

1. Осуществляет маршрутизацию запроса.
2. Подписывает ЭП Ядра ПОДД СМЭВ чанк метаданных запроса.
3. Передает в Агент Ответчика чанк метаданных или возвращает ошибку обработки запроса (при ее возникновении).
4. Получает подтверждение приема чанка метаданных от Агента Ответчика.
5. Передает подтверждение корректности запроса Агенту Инициатора.

*Агент Инициатора:*

1. Подписывает ЭП Инициатора чанки тела.
2. Передает последовательно чанки тела запроса в Ядро ПОДД СМЭВ с сохранением исходного порядка.

*Ядро ПОДД СМЭВ:*

1. Проверяет ЭП Инициатора каждого чанка тела запроса.
2. Подписывает ЭП Ядра ПОДД СМЭВ чанки тела запроса.
3. Последовательно передает чанки тела запроса в Агент Ответчика по протоколу RSocket.

*Агент Ответчика:*

1. Проверяет ЭП Ядра.
2. Проверяет ЭП Инициатора.
3. Выполняет проверку соответствия ограничениям, заданным на REST-сервис ИС Ответчика:

* на количество запросов от ИС Инициатора за интервал времени;
* на объем данных, передаваемых ИС Потребителя по всем запросам за интервал времени.

1. Формирует абсолютный URL-адрес запроса к соответствующему методу REST-сервиса ИС Ответчика.
2. Отправляет запрос данных согласно URL-адресу.

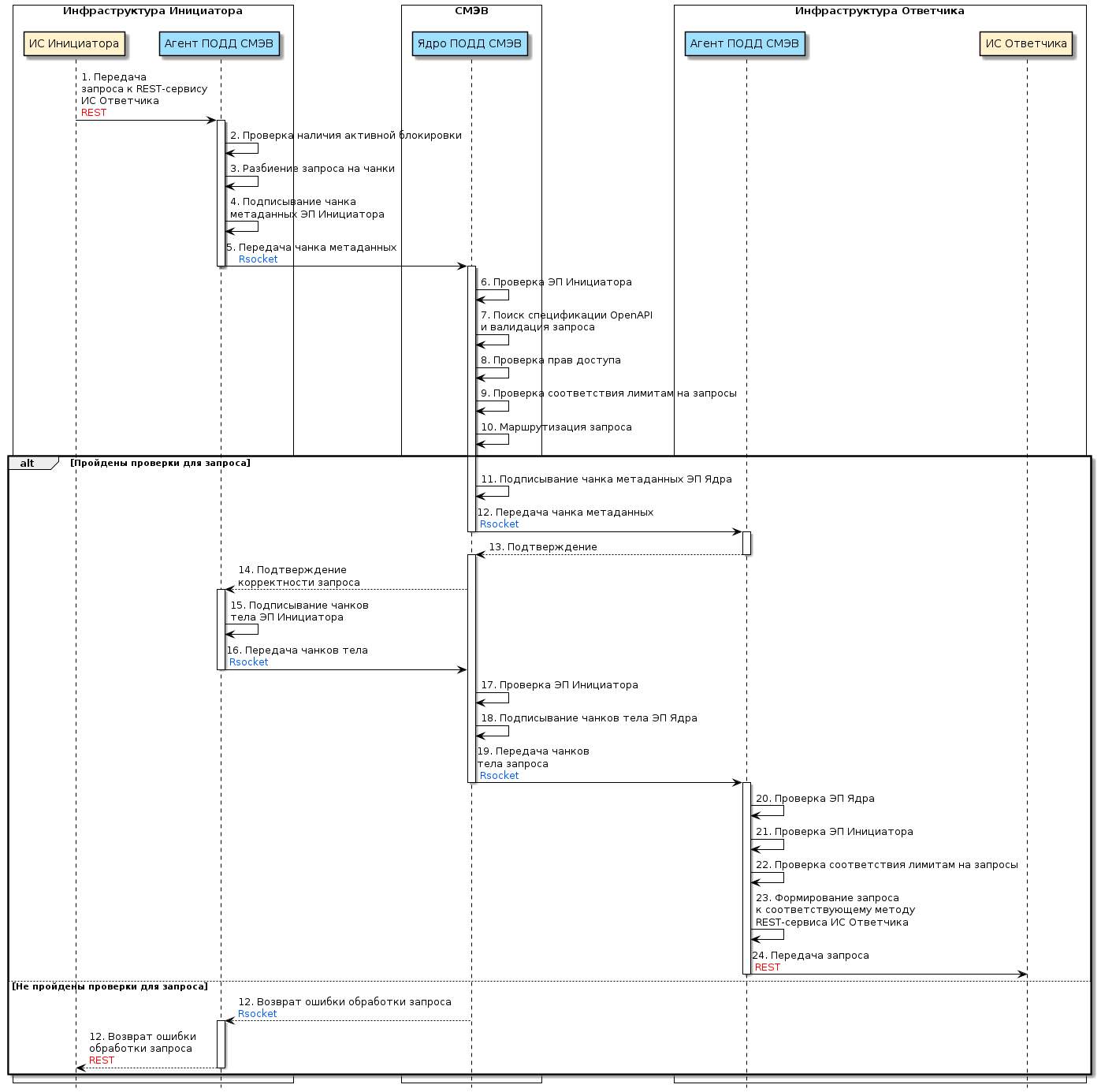


Рисунок - 1.11 Диаграмма прохождения запроса к REST-сервису ИС Ответчика через ПОДД СМЭВ

*Взаимодействие ИС Инициатора и ИС Ответчика при ответе на запрос к REST-сервису ИС Ответчика* ([Рисунок - 1.12](#_b58f04b1edd38d2829ffe2aa0202294e)):

*Агент Ответчика:*

1. Получает ответ на запрос от ИС Ответчика.
2. Делит ответ на запрос на чанк метаданных и чанки тела.
3. Подписывает чанк метаданных ЭП Ответчика.
4. Передает чанк метаданных в Ядро ПОДД СМЭВ.

*Ядро ПОДД СМЭВ:*

1. Проверяет ЭП Ответчика чанка метаданных.
2. Выполняет проверку соответствия ограничениям, заданным на REST-сервис ИС Ответчика, на объем данных, возвращаемых ИС Инициатора по всем запросам за интервал времени.
3. Маршрутизирует ответ на запрос.
4. Отправляет чанк метаданных в Агент Инициатора или возвращает ошибку обработки запроса (при ее возникновении).
5. Получает подтверждение приема чанка метаданных от Агента Инициатора.
6. Передает подтверждение корректности ответа на запрос в Агент Ответчика.

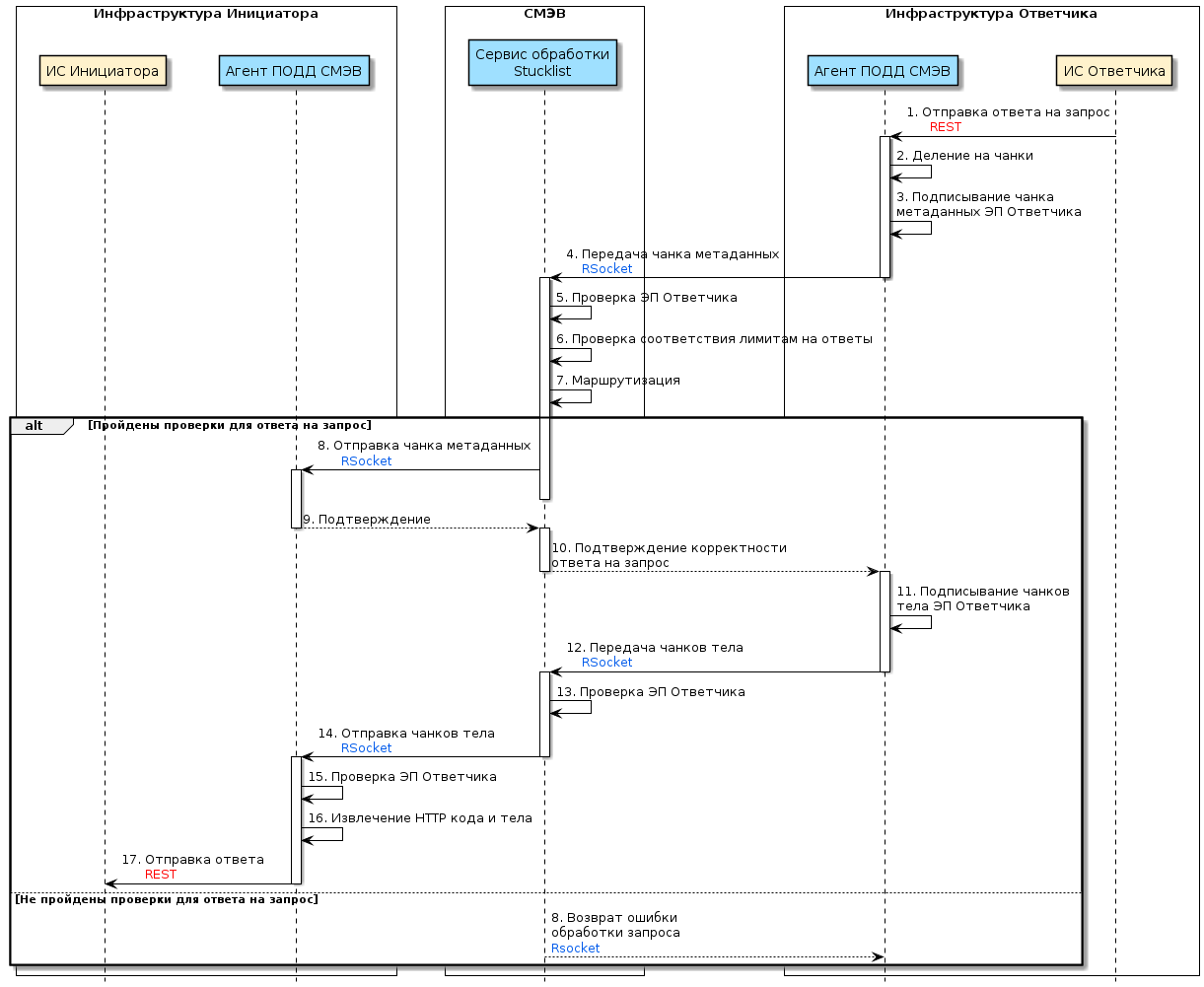


Рисунок - 1.12 Диаграмма прохождения ответа на запрос к REST-сервису ИС Ответчика через ПОДД СМЭВ

*Агент Ответчика:*

1. Подписывает ЭП Ответчика чанки тела.
2. Отправляет последовательно чанки тела в Ядро ПОДД СМЭВ.

*Ядро ПОДД СМЭВ:*

1. Проверяет ЭП Ответчика чанков тела.
2. Последовательно передает чанки тела в АгентИнициатора.

*Агент Инициатора:*

1. Выполняет проверку ЭП Ответчика.
2. Извлекает из ответа на запрос HTTP код и тело (если присутствует).
3. Отправляет ответ ИС Инициатора.

## 1.5 Метаданные ПОДД СМЭВ

Метаданные ПОДД СМЭВ содержат информацию о следующих объектах:

1. модели данных Витрин Поставщиков данных;
2. подписки Потребителей данных;
3. определения регламентированных SQL-запросов;
4. REST-сервисы ИС Ответчиков, описываемые спецификацией OpenAPI;
5. полномочия Потребителей данных.

### 

### 1.5.1 Поддерживаемые форматы

[Таблица 1.10](#_d09c3f4d8dab6e6917702c75e04cea8b) содержит ограничения на формат значений в загружаемых метаданных и сведения о проверках на стороне ПОДД СМЭВ.

Таблица 1.10 Ограничения на формат значений в метаданных

| № | Наименование | Тип | Требования к формату |
| --- | --- | --- | --- |
| 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 (ИС/Витрины/ регламентированного запроса типа «SQL-запрос»/ регламентированного запроса типа «Рассылка» /атрибута/параметра) | String | Латинские буквы, цифры и символ «\_», без пробелов. Первым символом должна быть буква. Регистр не учитывается |
| 3 | Мнемоника (Витрины/ регламентированного запроса типа «SQL-запрос» / регламентированного запроса типа «Рассылка») | String | Запрещено использование зарезервированных ключевых слов SQL [[11]](#footnote-11) |
| 4 | Мнемоника (регламентированного запроса типа «SQL-запрос» /регламентированного запроса типа «Рассылка») | String | Запрещено использование префикса information\_schema\_ |
| 5 | Алиас атрибута в SQL-запросе | String | Латинские буквы, цифры и символ «\_», без пробелов. Первым символом должна быть буква. Регистр не учитывается |
| 6 | Версия (Витрины/ регламентированного запроса типа «SQL-запроса»/ регламентированного запроса типа «Рассылка») | String | Должна соответствовать формату <major>.< minor >:   * major типа int, >= 0 * minor типа int, >= 0 |
| 7 | ОГРН УВ (владельца Витрины/ регламентированного запроа типа «SQL-запроса»/ регламентированного запроса типа «Рассылка») | String | ОГРН владельца Витрины должен соответствовать значению в ранее зарегистрированных версиях Витрины *Ожидаемый формат:* только цифры |
| 8 | Даты поддержки (Витрины/  регламентированного запроса типа «SQL-запрос»/ регламентированного запроса типа «Рассылка») | Integer  (int64) | В миллисекундах от эпохи.  Временной диапазон периода поддержки регламентированного запроса типа «SQL-запрос» и регламентированного запроса типа «Рассылка» должен входить в диапазоны поддержки всех Витрин, которые указаны в SQL данных регламентированного запроса типа «SQL-запрос» и регламентированного запроса типа «Рассылка» |
| 9 | Текстовое описание (Витрины/ регламентированного запроса типа «SQL-запроса»/ регламентированного запроса типа «Рассылка») | String | *Ожидаемый формат:* латинские и русские буквы, цифры, печатные символы и символы разметки (пробел, перенос строки), максимум 2000 знаков |
| 10 | Текст SQL-выражения (для регламентированного запроса типа «SQL-запрос»/ регламентированного запроса типа «Рассылка») | String | Поддерживаются возможности, указанные в [разделе 3.1](#_3.1_SQL-синтаксис) данного документа |
| 11 | Тип атрибута | String | * «BOOLEAN» * «STRING» * «INTEGER» * «LONG» * «FLOAT» * «DOUBLE» * «BIG\_DECIMAL» * «DATE» * «TIME» * «TIMESTAMP» * «BINARY» |
| 12 | Тип параметра | String | * «BOOLEAN» * «STRING» * «INTEGER» * «LONG» * «FLOAT» * «DOUBLE» * «BIG\_DECIMAL» * «DATE» * «TIME» * «TIMESTAMP» |
| 12 | Префикс в URL для REST-сервиса ИС Ответчика (basepath) | 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 „/“ |
| 13 | Спецификации OpenAPI для REST-сервиса ИС Ответчика |  | Поддерживаемые форматы: JSON, YAML. Максимальный размер 1 Мб |

### 1.5.2 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» в соответствии с инструкцией по работе с ФГИС «ЕИП НСУД»[[12]](#footnote-12).

**Модель данных Витрины Поставщика данных содержит:**

1. уникальную мнемонику Витрины;
2. ОГРН владельца Витрины;
3. версии Витрины, содержащие структуру сущностей.

**Каждая версия Витрины содержит:**

1. номер версии Витрины;
2. описание для версии Витрины;
3. срок действия версии Витрины Поставщика данных (дата начала и дата окончания действия версии Витрины);
4. перечень таблиц версии Витрины со следующей информацией:

* мнемоника и описание таблицы;
* перечень атрибутов таблицы с указанием типа данных;
* первичный ключ, состоящий из мнемоник атрибутов таблицы;
* массив уникальных ключей, состоящий из мнемоник атрибутов таблицы.

**Правила/ограничения использования:**

1. Модель данных Витрины Поставщика данных загружена в Ядро ПОДД СМЭВ, если загружена как минимум одна (первая) версия Витрины.
2. Значение ОГРН владельца должно быть одинаковым для всех версий Витрины. При несоответствии ОГРН значениям в ранее зарегистрированных версиях Витрины ПОДД вернет ошибку регистрации.
3. Актуальной версией Витрины считается максимальная по номеру среди действующих.
4. Если в новой версии Витрины требуется расширить перечень доступных данных и при этом Потребители, уже имеющие полный доступ к Витрине, не должны автоматически получить доступ к новой версии, то изменение следует выполнить через регистрацию новой Витрины (с новой мнемоникой), а не через выпуск новой ее версии.
5. Для выполнения запроса к определенной версии Витрины необходимо в запросе после мнемоники витрины явно указать номер версии. Пример приведен в [разделе 3.2.2](#_3.2.2_Указание_в) настоящего документа.

### 1.5.3 Подписки Потребителей данных ПОДД СМЭВ на регламентированный запрос типа «Рассылка»

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

[Таблица 1.11](#_12d4adf12fce02e70f0950f97158589b) содержит Метаданные подписки.

Таблица 1.11 Метаданные Подписки

| № | Атрибут | Тип | Обязательность | Описание |
| --- | --- | --- | --- | --- |
| 1 | subscriptionId | String  UUID | Да | Уникальный идентификатор подписки.  Генерируется ПОДД СМЭВ в соответствии с форматом UUID. |
| 2 | RQs | array of object | Да | Массив регламентированных запросов типа «Рассылка».  При указании нескольких запросов используется множественная подписка. Использование распределённых запросов в таком случае не допустимо. |
| 2.1 | queryMnemonic | String | Да | Полная мнемоника Регламентированного запроса: <мнемоника витрины>.<мнемоника РЗ>  Для распределенного запроса: podd.<мнемоника РЗ> |
| 2.2 | version | String | Да | Номер версии Регламентированного запроса в формате: <major>.<minor> |
| 3 | datamartMnemonic | String | Да | Мнемоника Витрины Потребителя (без учёта версионности) |
| 4 | name | String | Нет | Наименование подписки (для краткой формулировки) |
| 5 | description | String | Нет | Описание подписки (для подробного описания) |
| 6 | isSnapshot | Boolean | Нет | Признак того, что сначала Потребителю нужно выгрузить снапшот.  Значение по умолчанию - false |
| 7 | initiator | Boolean | Нет | Признак того, кто является инициатором запроса новой дельты:   * «PODD» - Ядро ПОДД СМЭВ; * «CONSUMER» – Потребитель (Ядро ПОДД СМЭВ передаёт уведомления о наличии дельт Потребителю, который инициирует запрос в соответствии с собственными настройками).   Значение по умолчанию - PODD |
| 8 | schedule | String | Нет | Расписание запроса Ядром ПОДД СМЭВ дельт у Витрины Поставщика.  Представляет собой cron-выражение.  Для подписок, в которых инициатором является Потребитель, значение атрибута всегда отсутствует. |
| 9 | status | String | Да | Статус подписки: формируется ПОДД СМЭВ по ходу регистрации и удаления подписки:   * REGISTRATION\_REQUEST – Подписка принята к регистрации в ПОДД; * REGISTRATION\_ON\_PRODUCER – Подписка регистрируется на Поставщике; * REGISTRATION\_ON\_PRODUCER\_ERROR – Подписка не зарегистрирована на Поставщике, возникла ошибка; * REGISTRATION\_ON\_CONSUMER – Подписка регистрируется на Потребителе; * REGISTRATION\_ON\_CONSUMER\_ERROR – Подписка не зарегистрирована на Потребителе, возникла ошибка; * REGISTERED – Подписка зарегистрирована; * CANCEL\_REQUEST – Принят запрос на удаление подписки (обмен по ней останавливается); * CANCEL\_ON\_PRODUCER (Удаление на Поставщике); * CANCEL\_ON\_CONSUMER (Удаление на Потребителе). |

Функционирование Агента СМЭВ4 и Ядра ПОДД СМЭВ при формировании первоначальной выгрузки и передачи изменений подписки от Витрины Поставщика данных в Хранилище данных по подписке приведено в [разделе 1.4.3.1](#_1.4.3.1_Общее_описание) и [разделе 2.4](#_2.4_Протокол_взаимодействия).

Регистрация подписки осуществляется через:

* ЛК УВ;
* ЕИП в случае тиражирования подписок (создание копии подписки, в которой изменяется только идентификатор и витрина потребителя).
* СЭ в случае необходимости ручного вмешательства.

Удаление подписки осуществляется через:

* ЛК УВ;
* СЭ в случае необходимости ручного вмешательства.

### 1.5.4 Определения регламентированных SQL-запросов

Регистрация регламентированного SQL-запроса в ПОДД СМЭВ осуществляется с использованием ЕИП НСУД в соответствии с инструкцией по работе с ФГИС «ЕИП НСУД»[[13]](#footnote-13).

**Состав атрибутов определения регламентированного SQL-запроса:**

1. мнемоника регламентированного SQL-запроса;
2. версии регламентированного SQL-запроса.

**Каждая версия регламентированного SQL-запроса содержит:**

1. описание регламентированного SQL-запроса (краткое и полное);
2. ОГРН владельца регламентированного SQL-запроса;
3. номер версии регламентированного SQL-запроса;
4. срок действия регламентированного SQL-запроса (дата начала и дата окончания действия, в рамках которых возможно выполнение соответствующего запроса);
5. SQL-выражение в соответствии с [разделом 3.1](#_3.1_SQL-синтаксис) и указанными ниже ограничениями;
6. (при наличии) список параметров. Описание параметра включает в себя:

* имя параметра;
* тип параметра;
* описание параметра;
* значение по умолчанию.

1. (при наличии) список табличных параметров. Описание параметров включает в себя:

* имя табличного параметра;
* описание колонок табличного параметра:
  + имя колонки
  + тип колонки;
  + описание колонки.

**Правила/ограничения использования:**

1. Версия регламентированного SQL-запроса не подлежит модификации. При необходимости изменить запрос создается запрос с новой версией.
2. ОГРН Владельца и мнемоника регламентированного SQL-запроса задаются для каждой версии отдельно, поэтому контроль их соответствия для версий ложится на создателей этих версий.
3. Актуальной версией регламентированного SQL-запроса считается максимальная по номеру среди действующих.
4. Если в новой версии регламентированного SQL-запроса требуется изменить входные или выходные данные и при этом Потребители, уже имеющие доступ к регламентированному SQL-запросу, не должны автоматически получить доступ к новой версии, то изменение следует выполнить через регистрацию нового регламентированного SQL-запроса (с новой мнемоникой), а не через выпуск новой его версии.
5. Временной диапазон срока действия регламентированного SQL-запроса должен входить в диапазоны поддержки всех Витрин, которые указаны в SQL-выражении данного Регламентированного SQL-запроса.
6. Параметры регламентированного SQL-запроса должны соответствовать условиям:

* табличные и именованные параметры в SQL-выражении должны соответствовать параметрам в списке;
* мнемоники параметров должны быть уникальными в рамках регламентированного SQL-запроса, уникальность именованных и табличных параметров сквозная;
* мнемоника параметра не должна начинаться с settings\_ (префикс системного именованного параметра);
* при задании параметров в SQL-выражении через «?» порядок параметров в списке должен соответствовать порядку их представления в SQL-выражении;
* при задании параметров в SQL-выражении через «?» количество параметров в SQL-выражении должно соответствовать количеству параметров в списке;

1. SQL-выражение для регламентированного SQL-запроса должно соответствовать условиям:

* Витрины, имена возвращаемых атрибутов, таблицы и схема данных должны соответствовать метаданным Витрины, сохраненным в Ядре ПОДД;
* должна быть указана конкретная версия Витрины (правильно – dtm.1.0.table; неправильно – dtm.table);
* указанная версия Витрины должна быть зарегистрирована в ПОДД СМЭВ;
* не должно быть указано несколько версий одной Витрины;
* простой запрос должен быть к одной Витрине;
* для простого запроса Витрина в SQL-выражении должна соответствовать Витрине, указанной в полной мнемонике Регламентированного SQL-запроса;
* распределенный запрос должен быть к двум или более Витринам;
* не должно быть обращений к другим Регламентированного SQL-запроса;
* в SQL-выражении должны быть указаны конкретные возвращаемые атрибуты (возврат всех атрибутов через \* запрещен). Исключение: возможно использовать \* внутри операторов (пример: count ( \* ));
* возвращаемые Потребителю атрибуты должны быть уникальны с учетом подставленных алисов.

1. Если регламентированный SQL-запрос планируется использовать в качестве Регламентированного запроса типа «Рассылка», то он должен соответствовать требованиям приведённым в [разделе 1.4.2.1](#_1.4.2.1_Общее_описание).

### 1.5.5 REST-сервисы ИС Ответчиков

Регистрация REST-сервиса ИС Ответчика (Регламентированного запроса типа REST-сервис) в ПОДД СМЭВ осуществляется с использованием ЛК УВ в соответствии с инструкцией в документе «Руководство пользователя ЛК УВ» [[14]](#footnote-14).

REST-сервис ИС Ответчика в ПОДД СМЭВ определяется следующими атрибутами:

1. Мнемоника ИС Ответчика.
2. Префикс в URL методов API (basepath).
3. Спецификация OpenAPI, описывающая REST-сервис ИС Ответчика.
4. Признак необходимости сквозной передачи подписи Инициатора Ответчику (по умолчанию true).

**Правила разработки спецификаций OpenAPI, используемых для описания и регистрации REST-сервисов Ответчиков в ПОДД СМЭВ:**

1. Следует руководствоваться форматом проектирования спецификаций OpenAPI версии 3.0 (в том числе минорных версий 3.0.1, 3.0.2 и т.д.).
2. В спецификациях OpenAPI могут быть использованы все типы HTTP-методов (POST, GET, PUT и DELETE), выполняющих CRUD-операции (Create/Read/Update/Delete).
3. Допустимые форматы: YAML или JSON.
4. Максимальный размер: 1 Мб.

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 Ответчика

{  
 "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": {}  
}

1. Спецификация OpenAPI, описывающая REST-сервис с функцией модификации данных ИС Ответчика:

{  
 "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"  
 ]  
 }  
 }  
 }  
}

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

Потребителю данных могут быть предоставлены следующие типы полномочий:

* полномочия на выполнение произвольных SQL-запросов;
* полномочия на выполнение регламентированных SQL-запросов;
* полномочия на выполнение запросов к REST-сервису ИС Ответчика;
* полномочия на Рассылку.

#### 1.5.6.1 Полномочия на выполнение произвольных SQL-запросов

Предоставляется автоматически и только владельцам Витрин данных на их собственные Витрины.

#### 1.5.6.2 Полномочия на выполнение регламентированных SQL-запросов

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

Полномочия могут быть предоставлены только для регламентированного SQL-запроса целиком. Права на атрибуты регламентированного SQL-запроса не выдаются.

При наличии полномочий на выполнение регламентированного SQL-запроса Потребитель данных автоматически получает доступ ко всем версиям этого запроса, включая будущие. Выдача полномочий в ПОДД СМЭВ осуществляется с использованием ЛК УВ в соответствии с инструкцией вдокументе «Руководство пользователя ЛК УВ» [[15]](#footnote-15).

Поставщику данных доступна донастройка полномочий на выполнение Регламентированного SQL-запроса с использованием Сервиса проверки полномочий, развернутого с Агентом СМЭВ4. Данный сервис обеспечивает возможность подтвердить или временно заблокировать полномочия на выполнение версии Регламентированного SQL-запроса, не изменяя загруженные в ПОДД СМЭВ полномочия.

#### 1.5.6.3 Полномочия на выполнение запросов к REST-сервису ИС Ответчика

Полномочия на выполнение запросов к REST-сервису ИС Ответчика (регламентированный запрос типа «REST-сервис») предоставляются на все запросы, указанные в загруженной в ПОДД СМЭВ спецификации OpenAPI для соответствующего REST-сервиса ИС Ответчика. Права на использование отдельных методов и запросов REST-сервиса не выдаются.

Управление полномочиями осуществляется с использованием ЛК УВ в соответствии с инструкцией в документе «Руководство пользователя ЛК УВ»[[16]](#footnote-16).

#### 1.5.6.4 Полномочия на Рассылку

Полномочия на доступ к регламентированному запросу типа «Рассылка» предоставляются исключительно органам исполнительной власти. Использование регламентированных запросов типа «Рассылка» кредитными организациями и в коммерческих целях запрещено. Управление полномочиями осуществляется через запрос в СЦ в соответствии с инструкцией «Правила и процедуры работы в подсистеме обеспечения доступа к данным федеральной государственной информационной системы «Единая система межведомственного электронного взаимодействия» по Методическим рекомендациям версии 4.XX»[[17]](#footnote-17).

## 

## 1.6 Типы данных ПОДД СМЭВ

Поддерживаемые ПОДД СМЭВ типы данных и соответствующие им физические и логические типы данных avro приведены в [таблице 1.12](#_2d8503779ccd24801a8ffc48b14b119d).

Типы данных приведены к используемым в системе Prostore (каноническое avro).

Таблица 1.12 Соответствие типов данных ПОДД СМЭВ и типов данных avro

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

# 2 Описание подключения к ПОДД СМЭВ

## 2.1 Подключение участников взаимодействия с использованием ПОДД СМЭВ

Основным способом направления обращений является использование Личного кабинета СЦ [https://sc.digital.gov.ru](https://sc.digital.gov.ru/).

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

Более подробная информация о порядке подключения к СМЭВ 4 приведена в документе «Правила и процедуры работы в подсистеме обеспечения доступа к данным федеральной государственной информационной системы «Единая система межведомственного электронного взаимодействия» по Методическим рекомендациям версии 4.XX»[[18]](#footnote-18).

## 2.2 Протокол взаимодействия Агента СМЭВ4 и Витрины Поставщика данных

Агент Поставщика данных может взаимодействовать с несколькими Витринами. Протокол коммуникации Агента СМЭВ4 и Витрин реализован в виде обмена сообщениями с использованием зарезервированных топиков брокера сообщений Apache Kafka.

### 2.2.1 Перечень топиков брокера сообщений Apache Kafka

Конфигурация Агента СМЭВ4 содержит перечень Витрин, с которыми он взаимодействует. Каждой Витрине в зависимости от настроек конфигурации, заданных в соответствии с документом «Руководством администратора ПОДД СМЭВ»[[19]](#footnote-19), соответствует один из наборов топиков:

1. Набор топиков, создаваемый по умолчанию. Полное название топиков формируются по шаблону <префикс для динамически регистрируемых Витрин>.<название топика>. По умолчанию префикс отсутствует.
2. Дополнительный набор топиков. Полное название топиков формируются по следующему шаблону <префикс для статически регистрируемых Витрин>.<название топика>. По умолчанию создаются отдельные группы топиков для каждой схемы Витрины с префиксом, соответствующим мнемонике Витрины.

Названия топиков брокера сообщений [[20]](#footnote-20) приведены в [таблице 2.1](#_7d016eac23d5c6b2b688ea149b50c7f8).

Таблица 2.1 Названия топиков брокера сообщений Apache Kafka

| № | Топик | Публикатор | Подписчик | Передаваемый объект |
| --- | --- | --- | --- | --- |
| *Топики регистрации и настройки Витрин* | | | | |
| 1 | datamart.signal | Витрина | Агент СМЭВ4 | Сообщение с регистрационными данными Витрины |
| 2 | <префикс>.profile.rq | Агент СМЭВ4 | Витрина | Запрос профиля Витрины |
| 3 | <префикс>.profile.rs | Витрина | Агент СМЭВ4 | Профиль Витрины |
| 4 | <префикс>.profile.err | Витрина | Агент СМЭВ4 | Ошибка при выполнении запроса профиля Витрины |
| 5 | <префикс>.signal | Агент СМЭВ4 | Витрина | Сообщение с итогами выполнения регистрации Витрины |
| *Топики для обеспечения информационного обмена с использованием SQL-запросов* | | | | |
| 6 | <префикс>.query.rq | Агент СМЭВ4 | Витрина | Подзапрос |
| 7 | <префикс>.procedure.query.rq | Агент СМЭВ4 | Витрина | Регламентированный запрос на исполнение |
| 8 | <префикс>.query.tp | Агент СМЭВ4 | Витрина | Подзапрос с использованием табличного параметра |
| 9 | <префикс>.query.tp.bin | Агент СМЭВ4 | Витрина | Подзапрос с использованием табличного параметра (при бинарном разбиении на чанки) |
| 10 | <префикс>.query.rs | Витрина | Агент СМЭВ4 | Результата подзапроса |
| 11 | <префикс>.query.estimation.rs | Витрина | Агент СМЭВ4 | Оценка (статистика) по исполнению подзапросов |
| 12 | <префикс>.query.err | Витрина | Агент СМЭВ4 | Ошибка при выполнении подзапроса |
| 13 | <префикс>.blob.rq | Агент СМЭВ4 | Витрина | Запрос двоичных данных по ссылке |
| 14 | <префикс>.blob.rs | Витрина | Агент СМЭВ4 | Результат запроса двоичных данных по ссылке |
| 15 | <префикс>.blob.err | Витрина | Агент СМЭВ4 | Ошибка при выполнении запроса двоичных данных по ссылке |
| 16 | <префикс>.cancel.rq | Агент СМЭВ4 | Витрина | Идентификатора запроса, выполнение которого в Витрине нужно отменить |
| 17 | <префикс>.cancel.rs | Витрина | Агент СМЭВ4 | Результат успешной отмены запроса |
| 18 | <префикс>.cancel.err | Витрина | Агент СМЭВ4 | Ошибка при выполнении отмены запроса |
| *Топики для обеспечения информационного обмена с использованием Рассылок* | | | | |
| 19 | <префикс>.replication.rq | Агент СМЭВ4 | Витрина | Информация о подписке |
| 20 | <префикс>.replication.rs | Витрина | Агент СМЭВ4 | Структура данных для хранения данных по подписке |
| 21 | <префикс>.replication.err | Витрина | Агент СМЭВ4 | Ошибка при обработке подписки |
| 22 | <префикс>.delta.rq | Агент СМЭВ4 | Витрина | Запрос пакета дельт по подписке |
| 23 | <префикс>.delta.tp | Агент СМЭВ4 | Витрина | Запрос на пересечение ключей и дельт по пересечённым ключам |
| 24 | <префикс>.delta.rs | Витрина | Агент СМЭВ4 | Дельта по подписке |
| 25 | <префикс>.delta.err | Витрина | Агент СМЭВ4 | Ошибка при формировании Витриной дельты по подписке |
| 26 | <префикс>.replication.cancel.rq | Агент | Витрина | Идентификатора отменяемой подписки |
| 27 | <префикс>.replication.cancel.rs | Витрина | Агент СМЭВ4 | Результат (успешный или ошибка) отмены подписки |
| 28 | <префикс>.delta.notification | Витрина | Агент СМЭВ4 | Уведомление о наличии дельты по подписке в Витрине Поставщика |
| *Топики для получения статистики по Витринам* | | | | |
| 29 | <префикс>.statistic.rq | Агент | Витрина | Запрос статистики таблиц |
| 30 | <префикс>.statistic.rs | Витрина | Агент СМЭВ4 | Статистика таблиц |
| 31 | <префикс>.statistic.err | Витрина | Агент СМЭВ4 | Ошибка формировании статистики таблиц |
| *Топики для получения событий Витрины* | | | | |
| 32 | <префикс>.scl.signal | Витрина | Агент СМЭВ4 | События Витрины для дальнейшей передачи в СЦЛ |
| *Топики для временного хранения сообщений при недоступности ПОДД СМЭВ* | | | | |
| 33 | <идентификатор ядра ПОДД СМЭВ>. <мнемоника агента>.undelivered. message | Агент СМЭВ4 | Агент СМЭВ4 | Сообщения от Витрины для передачи в ПОДД СМЭВ |

### 2.2.2 Настройка Агента СМЭВ4 для работы с несколькими Витринами

В ПОДД предусмотрено два способа регистрации Витрин:

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

Регистрация Витрины осуществляется вне зависимости от наличия загруженных в ПОДД СМЭВ метаданных Витрин (См. [Раздел 1.5.2](#_1.5.2_Модель_данных)). Выполнение запросов к Витринам возможно только после успешной регистрации Витрины и получения метаданных в ПОДД СМЭВ.

#### Регистрация на основе конфигурационного файла

Поддерживается на стороне Потребителя и Поставщика. [Таблица 2.2](#_7c82a6190ccc70634b8a28f85fe3ee79) содержит описание процесса регистрации Витрины.

Таблица 2.2 Процесс регистрации Витрины данных

| Процесс | Шаг | Описание |
| --- | --- | --- |
| **Инициация регистрации** | 1 | При запуске Агент СМЭВ4 вычитывает локальный конфигурационный файл [[21]](#footnote-21) с настройками регистрации Витрин. Витрина может быть зарегистрирована:   * статически (при явном указании в конфигурационном файле) * динамически (автоматическая регистрация Витрины при обращении к/от Витрины, без необходимости указания в конфигурационном файле)   и создает согласно настройкам в конфигурационном файле:   * группу топиков по умолчанию (для динамически регистрируемых Витрин) * одну или несколько дополнительных групп топиков (для статически регистрируемых Витрин) |
| 2 | Если одному из Агентов СМЭВ4 приходит сообщение из Ядра ПОДД СМЭВ для незарегистрированной Витрины, Агент СМЭВ4 регистрирует её динамически (и при необходимости создает группу топиков по умолчанию). |
| 3 | Если инициатором сообщения будет Витрина или ИС Потребителя, они будут зарегистрированы динамически, если при старте Агента СМЭВ4 создана группа топиков по умолчанию. |
| **Регистрация** | 4 | Агент СМЭВ4 загружает профиль Витрины. Способ загрузки определяется значением registrationFlow в конфигурационном файле:   * DEFINED\_PROFILE – загрузка по указанному в конфигурации пути (по умолчанию); * DATAMART\_REQUEST – запрос к Витрине через топик <префикс>.profile.rq |
| 5 | Витрина данных передаёт:   * либо профиль Витрины через топик <префикс>.profile.rs * либо сообщение об ошибке в топик <префикс>.profile.err |
| 6 | Агент СМЭВ4, получивший запрос от Витрины:   * уведомляет Ядро ПОДД СМЭВ о регистрации профиля новой Витрины, * передаёт в Ядро ПОДД СМЭВ профиль Витрины, * ожидает подтверждения регистрации от Ядра ПОДД СМЭВ |
| 7 | Ядро ПОДД СМЭВ: осуществляет регистрацию |
| 8 | Направляет в Агент СМЭВ4 подтверждение регистрации Витрины.  При успешном выполнении всех регистрационных операций либо если данная Витрина уже зарегистрирована возвращается статус «успех», в противном случае «не успех». |
| **Действия после успешной Регистрации** | 9 | Агент СМЭВ4 (при запуске) вычитывает метаданные Витрин из Ядра ПОДД СМЭВ через специальный топик.  Чтение метаданных не препятствует запуску, то есть в процессе чтения Агент СМЭВ4 уже может принимать запросы.  Обновление метаданных производится при каждом перезапуске Агента СМЭВ4 |
| **Действия после неуспешной Регистрации** | 10 | Агент СМЭВ4 завершает работу с ошибкой |

#### 2.2.2.2 Регистрация на основе сообщений Витрин

Только для Поставщика.

[Таблица 2.3](#_baf388f633a588a9d4e7b9f35f0a4e49) содержит описание процесса регистрации Витрины, схема регистрации на основе сообщения от Витрины приведена на [рисунке - 2.1](#_cc3337268ab2b77dfb33f9d8e261df2f).

Таблица 2.3 Процесс регистрации Витрины

| Процесс | Шаг | Описание |
| --- | --- | --- |
| **Инициация регистрации** | 1 | Витрина данных направляет в Агент СМЭВ4 сообщение с регистрационными данными в общий топик Kafka datamart.signal, предназначенный для получения запросов на регистрацию от всех Витрин.  При отправке запроса на регистрацию Витрина должна обеспечить подписку на топик signal |
| **Регистрация** | 2 | Агент СМЭВ4 создаёт необходимые для работы с новой Витриной данных топики в соответствии с [таблицей 2.1](#_7d016eac23d5c6b2b688ea149b50c7f8). |
| 3 | Агент СМЭВ4 запрашивает у Витрины профиль через топик profile.rq. |
| 4 | Агент СМЭВ4 получает:   * либо профиль Витрины через топик profile.rs * либо сообщение об ошибке в топик profile.err |
| 5 | Агент СМЭВ4:   * уведомляет Ядро ПОДД СМЭВ о регистрации профиля новой Витрины, * передаёт в Ядро ПОДД СМЭВ профиль Витрины, * ожидает подтверждения регистрации от Ядра ПОДД СМЭВ. |
| 6 | Агент СМЭВ4 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 «не успех». |
| 7 | Агент СМЭВ4 направляет Витрине сообщение о статусе регистрации. |
| 8 | В случае неуспешной регистрации Агент СМЭВ4 удаляет созданные на шаге 2 топики в Kafka. |
| **Действия после успешной Регистрации** | 9 | Агент СМЭВ4 выполняет запрос к Ядру ПОДД СМЭВ за актуальной информацией о зарегистрированных Витринах (метаданные Витрины и иные данные, необходимые для корректной работы)[[22]](#footnote-22) и обновляет по полученным данным хранимую локально сводную информацию.  Агент СМЭВ4 завершает процедуру запуска только после получения актуальных данных. |
| **Действия после неуспешной Регистрации** | 10 | В случае ошибок, таймаутов и сбоев, инициация повторной регистрации обеспечивается Витриной. |

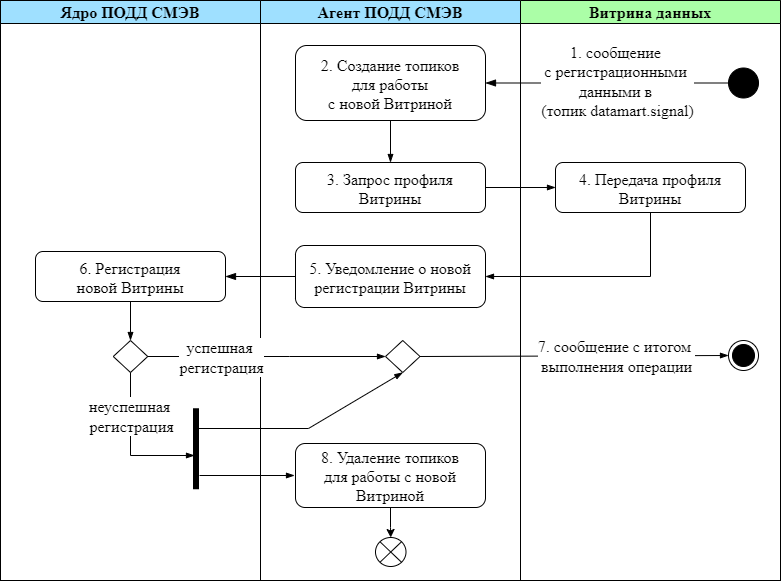


Рисунок - 2.1 Процесс регистрации Витрины данных на основе сообщения от Витрины данных

### 2.2.3 Примеры реализации взаимодействия с Агентом СМЭВ4 с использованием брокера сообщений Apache Kafka

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

* получение запроса от Агента СМЭВ4;
* передача результата Агенту СМЭВ4.

#### 2.2.3.1 Получение запроса от Агента СМЭВ4

Пример программного кода для получения запроса от Агента СМЭВ4 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()  
}

#### 2.2.3.2 Передача результата Агенту СМЭВ4

Пример программного кода для передачи результата выполнения запрос Агенту СМЭВ4 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 ?: ""  
 )  
  
}

## 2.3 Протокол взаимодействия Агента СМЭВ4 и ИС Потребителя данных

Для обеспечения доступности данных Агент СМЭВ4 предоставляет:

* REST-интерфейс для выполнения запросов к Витринам Поставщиков данных (См. [Раздел 2.3.1](#_2.3.1_REST-интерфейс_Агента));
* REST-интерфейс для выполнения запросов к REST-сервису ИС Ответчика (См. [Раздел 2.3.2](#_2.3.2_REST-интерфейс_Агента)).
* специализированный протокол для исполнения запросов к Витринам Поставщиков данных с использованием JDBC-интерфейса Агента СМЭВ4 (См. [Раздел 2.3.3](#_2.3.3_JDBC-интерфейс_Агента)).

### 2.3.1 REST-интерфейс Агента СМЭВ4 для SQL-запросов

В Агенте СМЭВ4 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 СМЭВ4;
* <порт> – порт, на котором развернут REST-интерфейс в соответствии с «Руководством администратора Агента ПОДД СМЭВ»[[23]](#footnote-23).

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 Base64.

Файлы в составе результата выполнения запроса BLOB по ссылке передаются в виде массива байт. Возможно выполнение запросов к Витринам Поставщиков данных:

* в синхронном режиме (см. [Раздел 2.3.1.1](#_2.3.1.1_Выполнение_SQL-запросов));
* в асинхронном режиме (см. [Раздел 2.3.1.2](#_2.3.1.2_Выполнение_SQL-запросов));
* на получение BLOB по ссылке (см. [Раздел 2.3.1.4](#_2.3.1.4_Выполнение_запроса)).

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

В синхронном режиме получение результата осуществляется путем выполнения HTTP-запроса от ИС Потребителя к Агенту СМЭВ4.

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

##### 2.3.1.1.1 HTTP-запрос

Таблица 2.4 HTTP-запрос

| № | Параметр | Тип | Обязательность | Описание |
| --- | --- | --- | --- | --- |
| 1 | Тип запроса (Method) |  | Да | POST |
| 2 | Путь (Path) |  | Да | <IP:port>/query |
| **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) |
| 3 | ClientRequestID | string | Нет | Клиентский идентификатор |
| **query** | | | | |
| 1 | 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.  Запросы без указания данного параметра использовать не рекомендуется, следует указать значение, равное времени ожидания ответа на ИС Потребителя. Максимальное значение 24 часа  В случае отсутствия параметра в запросе таймаут имеет значение по умолчанию 1 час |
| 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”}

##### 2.3.1.1.2 HTTP-запрос на вызов Регламентированного SQL-запроса без надстроек

Таблица 2.5 HTTP-запрос на вызов Регламентированного SQL-запроса без надстроек

| № | Параметр | Тип | Обязательность | Описание |
| --- | --- | --- | --- | --- |
| 1 | Тип запроса (Method) |  | Да | POST |
| 2 | Путь (Path) |  | Да | <IP:port>/regulated-query |
| **Заголовки** | | | | |
| 1 | Content-Type | string | Да | application/x-www-form-urlencoded; charset=utf-8 |
| 2 | Accept-version | string | Да | Основная (major) часть версии (сейчас 1) |
| 3 | ClientRequestID | string | Нет | Клиентский идентификатор |
| **Тело сообщения** | | | | |
| 1 | priority | string | Да | Приоритет запроса. Варианты:   * NORMAL; * HIGH |
| 2 | timeout | string | Нет | Предельное время ожидания выполнения запроса.  Запросы без указания данного параметра использовать не рекомендуется, следует указать значение, равное времени ожидания ответа на ИС Потребителя. Максимальное значение 24 часа  В случае отсутствия параметра в запросе таймаут имеет значение по умолчанию 1 час |
| 3 | datamart | string | Нет | Витрина Поставщика данных, к которой производится обращение |
| 4 | mnemonic | string | Нет | Мнемоника РЗ, к которому производится обращение |
| 5 | majorVersion | int | Нет  Если не указана majorVersion и minorVersion, обращение к актуальной версии | Версия РЗ, к которому производится обращение |
| 6 | majorVersion | int | Нет  Если не указана majorVersion и minorVersion, обращение к актуальной версии  Если не указана minorVersion и указанана majorVersion, обращение к последней версии в рамках majorVersion | Версия РЗ, к которому производится обращение |
| 7 | params | array | Нет | Параметры запроса |
| 7.1 | type | string | Да | Тип параметра |
| 7.2 | value | string | Да | Значение параметра |

Пример запроса:

POST «https://10.81.4.30:29354/regulated-query»  
Accept-Version:1  
Content-Type:application/x-www-form-urlencoded; encoding=utf-8  
  
priority:NORMAL  
timeout:60  
datamart: fias  
mnemonic: addrobj\_view  
majorVersion: 1  
minorVersion: 0

##### 2.3.1.1.3 Ответ на HTTP-запрос

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);
* 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 СМЭВ4);
* 503 – система временно недоступна, возможно повторить запрос через 50 мс.

Таблица 2.6 Параметры ответа с кодом возврата 200

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Параметр | Тип | Обязательность | Описание |
| 1 | responseCode | numeric | Да | Код возврата (HTTP-код) |
| **header** | | | | |
| 1 | Content-Type | string | Да | application/vnd.ru.rtlabs.podd.agent+json; charset=utf-8 |
| 2 | ClientRequestID | string | Нет | Клиентский идентификатор |
| 3 | Version | string | Да | Версия протокола (на текущий момент 1.0) |
| **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»  
 ]  
]  
}

Пример ответа при возврате BLOB и ссылки на BLOB:

HTTP/1.1 200 OK  
{  
 "created\_at": "2023-02-21T14:15:46Z",  
 "query\_id": "1edb1f23-90c1-6b75-bd1a-914f21e14802",  
 "meta": [  
 {  
 "name": "id",  
 "type": "INTEGER"  
 },  
 {  
 "name": "name",  
 "type": "STRING"  
 },  
 {  
 "name": "logo\_thumb",  
 "type": "BINARY"  
 }  
 ],  
 "rows": [  
При возврате BLOB:  
 [  
 "2",  
 "scala",  
 ""  
 ],  
При возврате ссылки на BLOB:  
 [  
 "17",  
 "test",  
 "link://c005a0e7-0d26-4ce0-a1fa-10c8bdf4dfc5"  
 ]  
 ]  
  
}

Таблица 2.7 Параметры ответа с кодом возврата, отличным от 200

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Параметр | Тип | Обязательность | Описание |
| 1 | responseCode | numeric | Да | Код возврата (HTTP-код) |
| **header** | | | | |
| 1 | Content-Type | string | Да | application/vnd.ru.rtlabs.podd.agent+json; charset=utf-8 |
| 2 | ClientRequestID | string | Нет | Клиентский идентификатор |
| 3 | Version | string | Да | Версия протокола (на текущий момент 1.0) |
| **body** | | | | |
| 1 | created\_at | dateTime | Да | Время формирования ответа |
| 2 | query\_id | string | Да | Идентификатор запроса |
| 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})’.”  
}

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

В асинхронном режиме получение результата осуществляется путем выполнения двух HTTP-запросов от ИС Потребителя к Агенту СМЭВ4:

1. В рамках первого HTTP-запроса (метод POST) передается SQL-запрос, в ответе возвращается идентификатор запроса.
2. 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.

##### 2.3.1.2.1 HTTP-запрос передачи SQL-запроса в Агент СМЭВ4 (метод POST)

###### 2.3.1.2.1.1 HTTP-запрос на вызов запроса с надстройками

Таблица 2.8 HTTP-запрос на вызов запроса с надстройками

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Параметр | Тип | Обязательность | Описание |
| 1 | Тип запроса (Method) |  | Да | POST |
| 2 | Путь (Path) |  | Да | <IP:port>/query |
| **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) |
| 3 | ClientRequestID | string | Нет | Клиентский идентификатор |
| **query** | | | | |
| 1 | async | boolean | Нет | Для асинхронного режима выполнения запроса параметр должен отсутствовать или иметь значение True |
| **body** | | | | |
| 1 | priority | string | Да | Приоритет запроса. Варианты:   * NORMAL; * HIGH |
| 2 | timeout | string | Нет | Предельное время ожидания выполнения запроса.  Запросы без указания данного параметра использовать не рекомендуется, следует указать значение, равное времени ожидания ответа на ИС Потребителя. Максимальное значение 24 часа  В случае отсутствия параметра в запросе таймаут имеет значение по умолчанию 1 час |
| 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”}

###### 2.3.1.2.1.2 HTTP-запрос на вызов запроса без надстроек

Таблица 2.9 HTTP-запрос на вызов запроса без надстроек

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Параметр | Тип | Обязательность | Описание |
| 1 | Тип запроса (Method) |  | Да | POST |
| 2 | Путь (Path) |  | Да | <IP:port>/regulated-query/async |
| **Заголовки** | | | | |
| 1 | Content-Type | string | Да | application/x-www-form-urlencoded; charset=utf-8 |
| 2 | Accept-version | string | Да | Основная (major) часть версии (сейчас 1) |
| 3 | ClientRequestID | string | Нет | Клиентский идентификатор |
| **Тело сообщения** | | | | |
| 1 | priority | string | Да | Приоритет запроса. Варианты:   * NORMAL; * HIGH |
| 2 | timeout | string | Нет | Предельное время ожидания выполнения запроса.  Запросы без указания данного параметра использовать не рекомендуется, следует указать значение, равное времени ожидания ответа на ИС Потребителя. Максимальное значение 24 часа  В случае отсутствия параметра в запросе таймаут имеет значение по умолчанию 1 час |
| 3 | datamart | string | Нет | Витрина Поставщика данных, к которой производится обращение |
| 4 | mnemonic | string | Нет | Мнемоника РЗ, к которому производится обращение |
| 5 | majorVersion | int | Нет  Если не указана majorVersion и minorVersion, обращение к актуальной версии | Версия РЗ, к которому производится обращение |
| 6 | majorVersion | int | Нет  Если не указана majorVersion и minorVersion, обращение к актуальной версии  Если не указана minorVersion и указанана majorVersion, обращение к последней версии в рамках majorVersion | Версия РЗ, к которому производится обращение |
| 7 | params | array | Нет | Параметры запроса |
| 7.1 | type | string | Да | Тип параметра |
| 7.2 | value | string | Да | Значение параметра |

Пример запроса:

POST «https://10.81.4.30:29354/regulated-query/async»  
Accept-Version:1  
Content-Type:application/x-www-form-urlencoded; encoding=utf-8  
  
priority:NORMAL  
timeout:60  
datamart: fias  
mnemonic: addrobj\_view  
majorVersion: 1  
minorVersion: 0  
params:{ “type”: “STRING”, “value”:”Москва”},{ “type”: “INTEGER”, “value”: “18”}

###### 2.3.1.2.1.3 Ответ на HTTP-запрос

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.

Таблица 2.10 Параметры ответа с кодом возврата 201:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Параметр | Тип | Обязательность | Описание |
| 1 | responseCode | numeric | Да | Код возврата (HTTP-код) |
| **header** | | | | |
| 1 | Content-Type | string | Да | application/vnd.ru.rtlabs.podd.agent+json; charset=utf-8 |
| 2 | Location | string | Да | Временная ссылка на скачивание результата выполнения запроса |
| 3 | ClientRequestID | string | Нет | Клиентский идентификатор |
| 4 | Version | string | Да | Версия протокола (на текущий момент 1.0) |
| **body** | | | | |
| 1 | id | string | Да | Уникальный идентификатор SQL-запроса |
| 2 | deadline | string | Да | Время, до которого доступен результат выполнения запроса. Время хранения результата составляет 24 часа |

Пример ответа с кодом возврата 201:

HTTP/1.1 201 Created  
{  
 “id”: “a2f05175-d5bc-47d4-9b88-17930630683e”,  
 «deadline»: «2021-05-13T06:33:43Z»  
}

Таблица 2.11 Параметры ответа с кодом возврата 400:

| № | Параметр | Тип | Обязательность | Описание |
| --- | --- | --- | --- | --- |
| 1 | responseCode | numeric | Да | Код возврата (HTTP-код) |
| **header** | | | | |
| 1 | Content-Type | string | Да | application/vnd.ru.rtlabs.podd.agent+json; charset=utf-8 |
| 2 | Version | string | Да | Версия протокола (на текущий момент 1.0) |
| **body** | | | | |
| 1 | error | string | Нет | Текст ошибки |

##### 2.3.1.2.2 HTTP-запрос получения результата, ранее переданного в Агент СМЭВ4 асинхронного SQL-запроса (метод GET)

###### 2.3.1.2.2.1 HTTP-запрос при вызове исходного запроса с надстройками

Таблица 2.12 HTTP-запрос при вызове исходного запроса с надстройками

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Параметр | Тип | Обязательность | Описание |
| 1 | Тип запроса (Method) |  | Да | GET |
| 2 | Путь (Path) |  | Да | <IP:port>/query |
| **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) |
| 3 | ClientRequestID | string | Нет | Клиентский идентификатор |
| **query** | | | | |
| 1 | query\_id | string | Да | Уникальный идентификатор SQL-запроса |

Пример запроса:

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

###### 2.3.1.2.2.2 HTTP-запрос при вызове исходного запроса без надстроек

Таблица 2.13 HTTP-запрос при вызове исходного запроса без надстроек

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Параметр | Тип | Обязательность | Описание |
| 1 | Тип запроса (Method) |  | Да | GET |
| 2 | Путь (Path) |  | Да | <IP:port>/regulated-query |
| **Параметры** | | | | |
| 1 | query\_id | string | Да | Уникальный идентификатор SQL-запроса |
| **Заголовки** | | | | |
| 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) |
| 3 | ClientRequestID | string | Нет | Клиентский идентификатор |

Пример запроса:

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

###### 2.3.1.2.2.3 Ответ на HTTP-запрос

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 СМЭВ4);
* 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.

Таблица 2.14 Параметры ответа с кодом возврата 200:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Параметр | Тип | Обязательность | Описание |
| 1 | responseCode | numeric | Да | Код возврата (HTTP-код) |
| **header** | | | | |
| 1 | Content-Type | string | Да | application/vnd.ru.rtlabs.podd.agent+json; charset=utf-8 |
| 2 | ClientRequestID | string | Нет | Клиентский идентификатор |
| 3 | Version | string | Да | Версия протокола (на текущий момент 1.0) |
| **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 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»  
 ]  
]  
}

Таблица 2.15 Параметры ответа с кодом возврата, отличным от 200:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Параметр | Тип | Обязательность | Описание |
| 1 | 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 | ClientRequestID | string | Нет | Клиентский идентификатор |
| **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 |

#### 2.3.1.3 Выполнение запроса с табличным параметром

Может быть выполнен как в синхронном, так и в асинхронном режиме. Ответ соответствует способу вызова.

##### 2.3.1.3.1 HTTP-запрос с табличным параметром на вызов запроса с надстройками

Таблица 2.16 HTTP-запрос с табличным параметром на вызов запроса с надстройками

| № | Параметр | Тип | Обязательность | Описание |
| --- | --- | --- | --- | --- |
| 1 | Тип запроса (Method) |  | Да | POST |
| 2 | Путь (Path) |  | Да | <IP:port>/query |
| **query** | | | | |
| 1 | async | boolean | Нет | Для синхронного режима выполнения запроса параметр должен отсутствовать или иметь значение False  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 |
| 4 | ClientRequestID | string | Нет | Клиентский идентификатор |
| **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 секундах  В случае отсутствия параметра в запросе таймаут не выставляется |
| 3 | sql | string | Да | Текст SQL-запроса |
| 4 | tableParams | array | Да | Описание передаваемого файла с данными для табличного параметра |
| 4.1 | name | string | Да | Наименование табличного параметра, соответствующее указанному в SQL-выражении в формате @<name> |
| 4.2 | columns | array | Да | Перечень наименований столбцов и их типов, содержащихся в файле с данными для табличного параметра |
| 4.2.1 | name | string | Да | Наименование столбца |
| 4.2.2 | type | string | Да | Тип столбца |
| 5 | <name> |  | Да | Файл с данными для табличного параметра. Имя параметра соответствует наименованию табличного параметра |
| **file** | | | | |
| 5.1 |  | файл в формате CSV | Да | Файл в формате CSV (поддерживаемый формат), передаваемый в параметре запроса |
| **boundary** | | | | |
| 1 | Content-Disposition | string | Да | form-data; name=”table1”; filename=”table1.csv” |

Пример запроса приведен в [разделе 3.3.4.1](#_3.3.4.1_Запрос_с) настоящего документа.

##### 2.3.1.3.2 HTTP-запрос с табличным параметром на вызов запроса без надстроек

Таблица 2.17 HTTP-запрос с табличным параметром на вызов запроса без надстроек

| № | Параметр | Тип | Обязательность | Описание |
| --- | --- | --- | --- | --- |
| 1 | Тип запроса (Method) |  | Да | POST |
| 2 | Путь (Path) |  | Да | <IP:port>/regulated-query – для синхронного вызова  <IP:port>/regulated-query/async – для асинхронного вызова |
| **Заголовки** | | | | |
| 1 | Content-Type | string | Да | multipart/form-data; |
| 2 | Connection | string | Да | keep-alive |
| 3 | Accept-version | string | Да | Основная (major) часть версии (сейчас 1) |
| 4 | ClientRequestID | string | Нет | Клиентский идентификатор |
| **Тело запроса** | | | | |
| 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 | datamart | string | Нет | Витрина Поставщика данных, к которой производится обращение |
| 4 | mnemonic | string | Да | Мнемоника РЗ, к которому производится обращение |
| 5 | majorVersion | int | Нет  Обязательное указание вместе с minorVersion. Если не указана, обращение к актуальной версии | Версия РЗ, к которому производится обращение |
| 6 | majorVersion | int | Нет  Обязательное указание вместе с majorVersion. Если не указана, обращение к актуальной версии | Версия РЗ, к которому производится обращение |
| 7 | tableParams | array | Да | Описание передаваемого файла с данными для табличного параметра |
| 7.1 | name | string | Да | Наименование табличного параметра,  соответствующее указанному в SQL-выражении в формате @<name> |
| 7.2 | columns | array | Да | Перечень наименований столбцов и их типов, содержащихся в файле с данными для табличного параметра |
| 7.2.1 | name | string | Да | Наименование столбца |
| 7.2.2 | type | string | Да | Тип столбца |
| 8 | <name> |  | Да | Файл с данными для табличного параметра. Имя параметра соответствует наименованию табличного параметра |
| **file** | | | | |
| 8.1 |  | файл в формате CSV | Да | Файл в формате CSV (поддерживаемый формат), передаваемый в параметре запроса |
| **boundary** | | | | |
| 1 | Content-Disposition | string | Да | form-data; name=”table1”; filename=”table1.csv” |

#### 2.3.1.4 Выполнение запроса на получение BLOB по ссылке

##### 2.3.1.4.1 HTTP-запрос (вариант 1)

Таблица 2.18 HTTP-запрос (вариант 1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Параметр | Тип | Обязательность | Значение / описание |
| 1 | Тип запроса (Method) |  | Да | GET |
| 2 | Путь (Path) |  | Да | <IP:port>/query/blob |
| **Параметры** | | | | |
| 1 | link | string | Да | Ссылка на вырузку BLOB |
| **Заголовки** | | | | |
| 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) |
| 3 | ClientRequestID | string | Нет | Клиентский идентификатор в формате UUID.  Потребителю рекомендуется в запросе BLOB по ссылке указать тот же сквозной идентификатор, что и при исходном запросе.  Ответственность за соответствие идентификаторов лежит на Потребителе. |

Пример запроса:

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

##### 2.3.1.4.2 HTTP-запрос (вариант 2)

Таблица 2.19 HTTP-запрос (вариант 2)

| № | Параметр | Тип | Обязательность | Значение / описание |
| --- | --- | --- | --- | --- |
| 1 | Тип запроса (Method) |  | Да | GET |
| 2 | Путь (Path) |  | Да | <IP:port>/regulated-query/blob |
| **Параметры** | | | | |
| 1 | link | string | Да | Ссылка на вырузку BLOB |
| **Заголовки** | | | | |
| 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) |
| 3 | ClientRequestID | string | Нет | Клиентский идентификатор в формате UUID.  Потребителю рекомендуется в запросе BLOB по ссылке указать тот же сквозной идентификатор, что и при исходном запросе.  Ответственность за соответствие идентификаторов лежит на Потребителе. |

Пример запроса:

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

##### 2.3.1.4.3 Ответ на HTTP-запрос

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 – получен запрос на обращение к недоступной ссылке на BLOB (истекло время жизни);
* 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 СМЭВ4);
* 503 – система временно недоступна, возможно повторить запрос через 50 мс.

Параметры ответа с кодом возврата 200:

Таблица 2.20 Параметры ответа с кодом возврата 200:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Параметр | Тип | Обязательность | Значение / описание |
| 1 | responseCode | numeric | Да | Код возврата (HTTP-код) |
| **Заголовки** | | | | |
| 1 | Content-Type | string | Да | application/octet-stream |
| 2 | Content-Length |  | Да | Размер тела объекта в байтах |
| 3 | ClientRequestID | string | Нет | Клиентский идентификатор |
| 4 | Version | string | Да | Версия протокола (на текущий момент 1.0) |
| **Тело сообщения** | | | | |
| 1 |  |  | Да | Содержимое полученного по ссылке файла (в виде массива байт) |

Пример ответа с кодом возврата 200:

HTTP/1.1 200 OK  
<Содержимое полученного по ссылке файла (в виде массива байт)>

Таблица 2.21 Параметры ответа с кодом возврата, отличным от 200:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Параметр | Тип | Обязательность | Описание |
| 1 | responseCode | numeric | Да | Код возврата (HTTP-код) |
| **Заголовки** | | | | |
| 1 | Content-Type | string | Да | application/vnd.ru.rtlabs.podd.agent+json; charset=utf-8 |
| 2 | Version | string | Да | Версия протокола (на текущий момент 1.0) |
| **Тело сообщения** | | | | |
| 1 | created\_at | dateTime | Нет | Время формирования ответа |
| 2 | query\_id | string | Нет | Идентификатор запроса |
| 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=<>, подробности: ‘Превышен лимит по количеству скачиваний по ссылке... ”  
}

### 2.3.2 REST-интерфейс Агента СМЭВ4 для запросов к REST-сервису ИС Ответчика

#### 2.3.2.1 HTTP-запрос

Все запросы выполняются в синхронном режиме в соответствии с загруженной в ПОДД СМЭВ спецификацией OpenApi REST-сервиса ИС Ответчика. Примеры спецификаций OpenAPI REST-сервисов ИС Ответчиков приведены в [разделе 1.5.5](#_1.5.5_REST-сервисы_ИС).

Получение результата осуществляется путем выполнения HTTP-запроса от ИС Инициатора к Агенту Ответчика. 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 Ответчиков.

Информация о формировании запроса приведена в [разделе 3.4](#_3.4_Выполнение_запросов).

Клиентский идентификатор передаётся в опциональном заголовке ClientRequestID (тип string).

#### 2.3.2.2 Ответ на HTTP-запрос

Ответ на запрос включает код возврата (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 Ответчика.

[Таблица 2.22](#_df1959749a0f2059f2bda6e124cdb44e) содержит коды возврата ПОДД при ошибках выполнения запроса к REST-сервису ИС Ответчика.

Таблица 2.22 Коды возврата ПОДД

|  |  |  |
| --- | --- | --- |
| № | Код возврата | Пример причины ошибки |
| 1 | 500 Internal Server Error | * ошибка пересылки сообщений между сервисами * непредвиденная ошибка |
| 2 | 401 Unauthorized | Ошибка проверки подписи |
| 3 | 404 Not found | Спецификация OpenAPI REST-сервиса ИС Ответчика, к которой обращается запрос, не зарегистрирована в ПОДД |
| 4 | 403 Forbidden | Нет прав на выполнение запроса к зарегистрированной в ПОДД спецификация OpenAPI REST-сервиса ИС Ответчика |
| 5 | 400 Bad request | Запрос не соответствует зарегистрированной в ПОДД спецификации OpenAPI REST-сервиса ИС Ответчика |
| 6 | 429 Too many requests | ИС УВ временно заблокирована в связи с превышением лимитов. Может возвращаться при использовании механизма с возможностью отправки большого запроса |
| 7 | 503 Service unavailable | Oтветчик ограничил очередь запросов в клиенте. Может возвращаться при использовании механизма без возможности отправки большого запроса |

### 2.3.3 JDBC-интерфейс Агента СМЭВ4 для SQL-запросов

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 в соответствии с «Руководством администратора Агента ПОДД СМЭВ»[[24]](#footnote-24).

Клиентский идентификатор опционально передаётся в тексте SQL запроса первой строкой – комментарий с атрибутом ClientRequestID (тип string).

-- ClientRequestID: <клиентский идентификатор>  
CALL <мнемоника регламентированного SQL-запроса>();

#### 2.3.3.1 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")  
 }  
 }  
 }  
 }  
 }  
 }  
 }

#### 2.3.3.2 Коды возврата

[Таблица 2.23](#_20d68cebeb3949c091f4f424b2a68a2c) содержит допустимые коды возврата при ошибках выполнения запроса.

Таблица 2.23 Коды возврата при ошибках выполнения запроса

|  |  |  |
| --- | --- | --- |
| № | Код возврата | Описание ошибки |
| 1 | 17001 | Внутренняя ошибка |
| 2 | 17473 | Запрос не прошел проверку корректности (соответствие синтаксису) |
| 3 | 17471 | ИС УВ временно заблокирована в связи с превышением лимитов |
| 4 | 17800 | Запрос содержит указание на неподдерживаемую Витрину |
| 5 | 17472 | Нет полномочий на выполнение запроса |
| 6 | 17510 | Запрос отменен Потребителем |
| 7 | 17520 | Запрос отменен по таймауту |
| 8 | 17404 | Выполнение запроса прекращено из-за блокировки полномочий по результатам проверки на стороне Поставщика |
| 9 | 17405 | Выполнение запроса прекращено из-за блокировки полномочий по результатам проверки на стороне Поставщика (заблокировано по умолчанию) |
| 10 | 17406 | Выполнение запроса прекращено из-за блокировки по результату проверки SQL выражения на стороне Поставщика |

## 2.4 Протокол взаимодействия Агента СМЭВ4 и Витрины по подписке Потребителя данных

Протокол коммуникации Агента СМЭВ4 и Витрины по подписке, расположенных в контуре Потребителя данных, устроен в виде обмена сообщениями с использованием зарезервированных топиков брокера сообщений Apache Kafka.

Всё взаимодействие между Витриной по подписке и Агентом СМЭВ4 происходит исключительно с использованием брокера сообщений.

### 2.4.1 Перечень топиков брокера сообщений Apache Kafka

[Таблица 2.24](#_359ffb9e1c4200168b5c892e521c4343) содержит названия топиков брокера сообщений.

Таблица 2.24 Названия топиков брокера сообщений Apache Kafka

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Топик | Публикатор | Подписчик | Передаваемый объект |
| **Топики для обеспечения информационного обмена с использованием Рассылок** | | | | |
| 1 | <префикс>.replication.in.rq | Агент СМЭВ4 | Витрина | Структура таблиц Витрины Поставщика данных |
| 2 | <префикс>.replication.in.rs | Витрина | Агент СМЭВ4 | Уведомление об успешном создании структуры данных |
| 3 | <префикс>.replication.in.err | Витрина | Агент СМЭВ4 | Уведомление об ошибке при создании структуры данных |
| 4 | <префикс>.delta.notification.in | Агент СМЭВ4 | Витрина | Уведомление о наличии новых дельт у Поставщика |
| 5 | <префикс>.command.podd | ИС Потребителя | Агент СМЭВ4 | Служебный топик для ручного перезапроса дельт |
| 6 | <префикс>.delta.in.rq | Агент СМЭВ4 | Витрина | Запрос на прием дельты |
| 7 | <префикс>.delta.in.tp | Агент СМЭВ4 | Витрина | Запрос на прием дельт по распределённой подписке |
| 8 | <префикс>.delta.in.rs | Витрина | Агент СМЭВ4 | Уведомления об успешном применении дельт из пакета |
| 9 | <префикс>.delta.in.err | Витрина | Агент СМЭВ4 | Уведомление об ошибке при применении дельт из пакета |
| 10 | <префикс>.replication.cancel.in.rq | Агент СМЭВ4 | Витрина | Идентификатора отменяемой подписки |
| 11 | <префикс>.replication.cancel.in.rs | Витрина | Агент СМЭВ4 | Результат (успешный или ошибка) отмены подписки |
| **Топики для получения событий Витрины** | | | | |
| 12 | <префикс>.scl.signal | Витрина | Агент СМЭВ4 | События Витрины для дальнейшей передачи в СЦЛ |

## 2.5 Протокол взаимодействия Агента СМЭВ4 и ИС Ответчика

Взаимодействие Агента СМЭВ4 и REST-сервиса на стороне ИС Ответчика осуществляется в соответствии со спецификацией OpenAPI, описывающей REST-сервис ИС Ответчика и загруженной в ПОДД СМЭВ.

Для использования запросов к REST-сервису ИС Ответчика через ПОДД СМЭВ необходимо произвести настройки, указанные в Руководстве администратора Агента СМЭВ4[[25]](#footnote-25).

# 3 Использование ПОДД СМЭВ

## 3.1 SQL-синтаксис

Регламентированные SQL-запросы, регистрируемые в ПОДД СМЭВ, должны соответствовать следующему синтаксису.

Таблица 3.1 SQL-синтаксис

|  |  |  |
| --- | --- | --- |
| № | 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 int\_column 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 int\_column 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 char\_column FROM t WHERE char\_column > varchar\_column |
| 17 | Функция POSITION | SELECT position(‘A’ IN char\_column) FROM t |
| 18 | Сравнения символов | SELECT char\_column 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 | Корреляционные имена в предложении FROM | SELECT K.column FROM t AS K |
| 28 | Переименование колонок в предложении FROM | SELECT column FROM t AS x(q, c) |
| **Базовые предикаты и условия поиска** | | |
| 29 | Предикат сравнения | SELECT column FROM t WHERE 0 = 0 |
| 30 | Предикат BETWEEN | SELECT column FROM t WHERE ‘ ‘ BETWEEN ‘’ AND ‘’ |
| 31 | Предикат IN со списком значений | SELECT column FROM t WHERE char\_column IN (‘a’, upper(‘a’)) |
| 32 | Предикат LIKE | SELECT column FROM t WHERE char\_column LIKE ‘\_’ |
| 33 | Предложение ESCAPE в предикате LIKE | SELECT column FROM t WHERE ‘abc’ LIKE ‘abcX\_’ ESCAPE ‘X’ |
| 34 | Предикат NULL | SELECT column FROM t WHERE char\_column IS NOT NULL |
| 35 | Предикаты количественного сравнения | SELECT column FROM t WHERE char\_column = ANY (SELECT char\_column FROM t) |
| 36 | Предикат EXISTS | SELECT column FROM t WHERE NOT EXISTS (SELECT char\_column FROM t) |
| 37 | Подзапросы в предикате сравнения | SELECT column FROM t WHERE int\_column > (SELECT max (int\_column) FROM t) |
| 38 | Подзапросы в предикате IN | SELECT column FROM t WHERE char\_column IN (SELECT char\_column FROM t) |
| 39 | Подзапросы в предикате количественного сравнения | SELECT column FROM t WHERE char\_column >= ALL(SELECT char\_column FROM t) |
| 40 | Коррелирующие подзапросы | SELECT column FROM t WHERE int\_column = (SELECT int\_column FROM t2 WHERE t2.char\_ column = t.char\_column) |
| 41 | Условие поиска | SELECT column FROM t WHERE 0 <> 0 OR ‘a’ < ‘b’ AND int\_column IS NOT NULL |
| **Простые выражения с запросами** | | |
| 42 | Табличный оператор UNION DISTINCT | SELECT column FROM t UNION DISTINCT SELECT column1 FROM t |
| 43 | Табличный оператор UNION ALL | SELECT column FROM t UNION ALL SELECT column1 FROM t |
| 44 | Табличный оператор EXCEPT DISTINCT | SELECT column FROM t EXCEPT DISTINCT SELECT column1 FROM t |
| 45 | Колонки, объединяемые табличными операторами, могут иметь разные типы данных | SELECT char\_column FROM t UNION SELECT 5 |
| 46 | Табличные операторы в подзапросах | SELECT column FROM t WHERE ‘a’ IN (SELECT char\_column FROM t UNION SELECT char\_column FROM t) |
| **Функции множеств** | | |
| 47 | AVG | SELECT avg(int\_column) FROM t |
| 48 | COUNT | SELECT count(int\_column) FROM t |
| 49 | MAX | SELECT max(int\_column) FROM t |
| 50 | MIN | SELECT min(int\_column) FROM t |
| 51 | SUM | SELECT sum(int\_column) FROM t |
| 52 | Дополнение ALL | SELECT sum(ALL int\_column) FROM t |
| 53 | Дополнение DISTINCT | SELECT sum(DISTINCT int\_column) FROM t |
| 54 | Оператор SELECT, возвращающий одну строку | SELECT count(\*) FROM t |
| **Базовая поддержка курсоров** | | |
| 55 | Колонки ORDER BY, отсутствующие в списке выборки | SELECT int\_column FROM t ORDER BY char\_column |
| 56 | Выражения значений в предложении ORDER BY | SELECT int\_column FROM t ORDER BY -int\_column |
| 57 | Поддержка NULL (NULL вместо значений) | SELECT int\_column FROM t WHERE int\_column IS NULL |
| **Базовое соединение таблиц** | | |
| 58 | 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 |
| 59 | Ключевое слово INNER | SELECT a.int\_column FROM t a JOIN t b ON a.int\_column = b.int\_column |
| 60 | 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 |
| 61 | 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 |
| 62 | 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 |
| 63 | 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 |
| 64 | Поддерживаются все операторы сравнения (а не только =) | SELECT column FROM t WHERE 0 = 1 OR 0 > 1 OR 0 < 1 OR 0 <> 1 |
| **Базовая поддержка даты и времени** | | |
| 65 | Тип данных DATE (включая поддержку строк DATE) | SELECT ‘2012-07-12’ AS "DATE" |
| 66 | Тип данных TIME (включая поддержку строк TIME) с точностью до секунд как минимум с 0 знаков после запятой | SELECT ‘1:2:3’ AS "TIME" |
| 67 | 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" |
| 68 | Предикаты сравнения с типами данных DATE, TIME и TIMESTAMP | SELECT column FROM t3 WHERE date\_column = date\_column AND time\_column = time\_column AND timestamp\_column = timestamp\_column |
| 69 | Явное приведение (CAST) между типами даты/времени и типами символьных строк | SELECT cast(date\_column AS VARCHAR(10)) FROM t3 |
| 70 | CURRENT\_DATE | SELECT current\_date FROM t |
| 71 | LOCALTIME | SELECT localtime FROM t |
| 72 | LOCALTIMESTAMP | SELECT localtimestamp FROM t |
| **Расширенная поддержка даты и времени** | | |
| 73 | FOR SYSTEM\_TIME (запрос данных, актуальных на указанную дату и время) | SELECT column FROM t FOR SYSTEM\_TIME AS OF 'YYYY-MM-DD hh:mm:ss'  SELECT column FROM t FOR SYSTEM\_TIME AS OF TIMESTAMP 'YYYY-MM-DD hh:mm:ss' |
| 74 | Указание «TIMESTAMP» опционально Вычисление интервала (c указанием единиц времени: YEAR, MONTH, DAY, HOUR, MINUTE, SECOND) | select ('YYYY-MM-DD hh:mm:ss' - 'YYYY-MM-DD hh:mm:ss') MONTH |
| 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 column FROM t WHERE row(int\_column, int\_column) IS NOT NULL |

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

Далее приведены примеры SQL-выражений регламентированных SQL-запросов.

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

### 3.2.1 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**

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

### 3.2.2 Указание в запросе версии Витрины

select \* **from** **egrul.**1.0.legalentity  
select \* **from** **egrul.**2.0.legalentity

Результат может содержать разное количество возвращаемых атрибутов в соответствии со структурой таблицы «legalentity» Витрины «egrul» версий 1.0 и 2.0.

### 3.2.3 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

## 3.3 Выполнение регламентированных SQL-запросов

### 3.3.1 Правила выполнения

Регламентированный SQL-запрос выполняется аналогично выполнению любого запроса к Витрине Поставщика данных. Описание синтаксиса выполнения запроса приведено в [Разделе 3.1](#_3.1_SQL-синтаксис) .

Эта операция может быть выполнена:

* с использованием REST-интерфейса в соответствии с [разделом 2.3.1](#_2.3.1_REST-интерфейс_Агента);
* с использованием JDBC-интерфейса в соответствии с [разделом 2.3.3](#_2.3.3_JDBC-интерфейс_Агента) .

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

select \* **from** <мнемоника Витрины>.<версия Регламентированного SQL-запроса>.<мнемоника Регламентированного SQL-запроса>(<параметры>) <дополнительные функции (опционально)>

где:

* <мнемоника Витрины> – для простых запросов (к одной Витрине) задается мнемоника соответствующей Витрины. Для распределенного запроса вместо мнемоники Витрины указывается префикс «podd»;
* <версия Регламентированного SQL-запроса> – задается в формате «major.minor»;
* <мнемоника Регламентированного SQL-запроса> – мнемоника вызываемого Регламентированного SQL-запроса;
* <параметры> – указываются в скобках через запятую для Регламентированного SQL-запроса с параметрами, заданными в определении.

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.

* <дополнительные функции (опционально)> – дополнительные функции для обработки вызываемого Регламентированного SQL-запроса (например, группировка или лимит).

|  |
| --- |
| Внимание: |
| если запрос не является аналитическим и необходимо получение результата выполнения в реальном времени (запросы граждан на портале ЕПГУ, предоставление услуг в МФЦ и т.п.), то обязательно указание лимита (limit) с минимально возможным значением (максимально допустимое значение зависит от конфигурации и настроек Витрины Поставщика, по умолчанию равно 1000):  POST «https://<host>:<port>/query?async=false»  Accept-Version:1 Content-Type:application/x-www-form-urlencoded; encoding=utf-8  priority:NORMAL timeout:60 sql: select \* from fias.1.0.addrobj\_view('001') limit 10 |

Для упрощенного вызова Регламентированного SQL-запроса без использования дополнительных функций обработки (без надстроек) доступен формат вызова:

* через REST-интерфейс в соответствии c [разделом 2.3.1.1.2](#_2.3.1.1.2_HTTP-запрос_на) с соответствующей пометкой;
* через JDBC-интерфейс:

CALL <мнемоника Витрины>.<версия Регламентированного SQL-запроса>.<мнемоника Регламентированного SQL-запроса>(<параметры>)

**Правила указания параметров:**

1. Если параметры заданы в SQL-выражении Регламентированного SQL-запроса в виде «?», то при вызове они должны быть указаны в порядке, соответствующем определению регламентированного SQL-запроса. Количество параметров при вызове должно соответствовать количеству параметров в определении Регламентированного SQL-запроса.
2. Для запроса, в SQL-выражении которого параметры заданы как именованные, доступен:
3. Вызов с использованием мнемоники именованного параметра.

Для этого необходимо задать значения всех именованных параметров, для которых не задано значение по умолчанию, в виде «<мнемоника параметра>=><значение параметра>» (СМ. [Раздел 3.3.2](#_3.3.2_Примеры_вызова)).

Мнемоника именованного параметра при вызове запроса должна соответствовать мнемонике именованного параметра в определении Регламентированного SQL-запроса.

Использование именованных параметров позволяет при вызове задавать значения параметров в любом порядке и только один раз, независимо от количества мест, где этот параметр используется в определении Регламентированного SQL-запроса.

1. Вызов без использования мнемоники именованного параметра.

Для этого необходимо задать значения всех именованных параметров в порядке, соответствующем определению Регламентированного SQL-запроса.

1. Совместное использование именованных и неименованных параметров не допускается.

### 3.3.2 Примеры вызова и преобразования исходного запроса

1. Регламентированный SQL-запрос с параметрами без имени:

// Исходный запрос от Потребителя данных ПОДД СМЭВ:  
select \* from oktmo.1.0.oktmo\_view(‘Московская область’,7)  
  
// Определение Регламентированного SQL-запроса  
select id, whenadd, name, regionname, settlementtypename FROM oktmo.1.0.oktmo where regionname = ? AND settlementtypeid = ?  
  
// Пример преобразования исходного запроса в соответствии с загруженным определением Регламентированного SQL-запроса:  
select id, whenadd, name, regionname, settlementtypename FROM oktmo.1.0.oktmo where regionname = ‘Московская область’ AND settlementtypeid = 7

1. Регламентированный SQL-запрос с именованными параметрами:

// Исходный запрос от Потребителя данных ПОДД СМЭВ (вызов через мнемонику параметра)  
select \* **from** <мнемоника Витрины>.<версия Регламентированного SQL-запроса>.<мнемоника Регламентированного SQL-запроса>(param=>'000', lim=>20)  
  
// Исходный запрос от Потребителя данных ПОДД СМЭВ  
select \* **from** <мнемоника Витрины>.<версия Регламентированного SQL-запроса>.<мнемоника Регламентированного SQL-запроса>('000',20)  
  
// Определение Регламентированного SQL-запроса  
SELECT name, code FROM datamart.1.0.regions WHERE code=:param **and** id **in** (select id **from** :param) limit :lim  
  
// Пример преобразования исходного запроса в соответствии с загруженным определением Регламентированного SQL-запроса:  
SELECT name, code FROM datamart.1.0.regions WHERE code='000' **and** id **in** (select id **from** '000') limit 20

1. Регламентированный SQL-запрос с параметром типа TIMESTAMP:

// Исходный запрос от Потребителя данных ПОДД СМЭВ при подключении к Pulsar  
select \* from fias.1.0.addrobj\_view('001', '000', TIMESTAMP '1970-01-01 00:00:00')  
  
// Исходный запрос от Потребителя данных ПОДД СМЭВ при подключении к брокеру  
CALL fias.1.0.addrobj\_view('001', '000', TIMESTAMP '1970-01-01 00:00:00')  
  
// Исходный запрос от Потребителя данных ПОДД СМЭВ при подключении к брокеру с отдельным блоком параметров  
datamart: fias  
mnemonic: addrobj\_view  
majorVersion: 1  
minorVersion: 0  
params:{ “type”: “STRING”, “value”:”001”},{ “type”: “STRING”, “value”:“002”},{ “type”: “TIMESTAMP”, “value”:“2007-12-03Т10:00:00.000Z”}  
  
// Определение РЗ  
SELECT oktmo, formalname, startdate, enddate FROM datamart.9.1.addrobj WHERE areacode=? and citycode=? and startdate<=?

1. Регламентированный SQL-запрос с параметром типа DATE:

// Исходный запрос от Потребителя данных ПОДД СМЭВ при подключении к Pulsar  
select \* from fias.1.0.addrobj\_view('001', '000', '1970-01-01')  
  
// Исходный запрос от Потребителя данных ПОДД СМЭВ при подключении к брокеру  
CALL fias.1.0.addrobj\_view('001', '000', '1970-01-01')  
  
// Исходный запрос от Потребителя данных ПОДД СМЭВ при подключении к брокеру с отдельным блоком параметров  
datamart: fias  
mnemonic: addrobj\_view  
majorVersion: 1  
minorVersion: 0  
params:{ “type”: “STRING”, “value”:”001”},{ “type”: “STRING”, “value”:“002”},{ “type”: “DATE”, “value”:“2007-12-03”}  
  
// Определение РЗ  
SELECT oktmo, formalname, startdate, enddate FROM datamart.9.1.addrobj WHERE areacode=? and citycode=? and startdate>=?

1. Регламентированный SQL-запрос 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 Регламентированного SQL-запроса:  
select ogrn, short\_name, inn, kpp, region\_code **from** **egrul.**2.legalentity

1. Распределенный Регламентированный SQL-запрос:

// 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 Регламентированного SQL-запроса:  
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

### 3.3.3 Выполнение запроса с системным параметром

Системные параметры не описываются в определении регламентированного SQL-запроса и могут быть заданы Потребителем для любого регламентированного SQL-запроса при выполнении запроса.

[Таблица 3.2](#_96e19940d46c2450c9e3881422f4208d) содержит поддерживаемые в ПОДД СМЭВ параметры.

Таблица 3.2 Поддерживаемые в ПОДД СМЭВ параметры

|  |  |  |
| --- | --- | --- |
| № | Мнемоника | Назначение |
| 1 | settings\_for\_system\_time | Системный параметр для получения актуальных на заданный момент данных.  Позволяет указывать при вызове запроса момент времени, на который требуется получить актуальные данные из Витрины. |
| 2 | settings\_for\_system\_time\_started | Системный параметр для получения результата выполнения запроса с данными, добавленными за указанный диапазон времени.  Позволяет указывать при вызове запроса диапазон номеров дельт, за который требуется получить измененные данные |
| 3 | settings\_for\_system\_time\_finished | Системный параметр для получения результата выполнения запроса с данными, удаленными за указанный диапазон времени.  Позволяет указывать при вызове запроса диапазон номеров дельт, за который требуется получить удаленные данные |

Вызов Регламентированного SQL-запроса с системным параметром осуществляется аналогично вызову Регламентированного SQL-запроса с заданием значения для именованного параметра, в соответствии со следующими правилами:

1. Параметр задается в запросе только один раз в виде именованного параметра с соответствующей мнемоникой.
2. Значение параметра settings\_for\_system\_time должно соответствовать формату YYYY-MM-DD hh:mm:ss.
3. Значение параметров settings\_for\_system\_time\_started и settings\_for\_system\_time\_finished должно соответствовать формату (int1, int2). Значение int1 должно быть меньше, чем int2.
4. Задание параметра недоступно при вызове регламентированного SQL-запроса без SQL-выражения.
5. Для простого запроса: для Витрины, к которой обращается регламентированный SQL-запрос с данным параметром, должна быть задана настройка поддержки обработки данного параметра в соответствии с документом «Руководство администратора Агента ПОДД СМЭВ».[[26]](#footnote-26)
6. Для распределенного запроса: для всех Витрин, к которым обращается регламентированный SQL-запрос с данным параметром, должна быть задана настройка поддержки обработки данного параметра в соответствии с документом «Руководство администратора Агента ПОДД СМЭВ». [[27]](#footnote-27)
7. В подзапросе в блоке FROM должна отсутствовать конструкция «FOR SYSTEM\_TIME».

Пример вызова с параметром settings\_for\_system\_time:

// вызываемый SQL-запрос (вариант 1)  
select \* from fias.1.0.addrobj\_func(p1=>'000', p2=>'001', settings\_for\_system\_time=>'YYYY-MM-DD hh:mm:ss')  
  
// вызываемый SQL-запрос (вариант 2)  
select \* from fias.1.0.addrobj\_func(settings\_for\_system\_time=>?, p1=>?, p2=>?)  
params: {"type": "TIMESTAMP", "value": "YYYY-MM-DD hh:mm:ss"},{"type": "STRING", "value": "001"},{"type": "STRING", "value": "002"}  
  
// SQL-выражение Регламентированного SQL-запроса  
SELECT <атрибуты> FROM <витрина>.<таблица> WHERE id=:p1 AND region=:p2

Пример вызова с параметрами settings\_for\_system\_time\_started» и «settings\_for\_system\_time\_finished:

// вызываемый SQL-запрос (вариант 1)  
select \* from fias.1.0.addrobj\_func(p1=>'000', p2=>'001', settings\_for\_system\_time\_started=>'122,124')  
select \* from fias.1.0.addrobj\_func(p1=>'000', p2=>'001', settings\_for\_system\_time\_finished=>'122,124')  
  
// вызываемый SQL-запрос без настроек через JDBC-интерфейс  
call fias.1.0.addrobj\_func(p1=>'000', p2=>'001', settings\_for\_system\_time\_started=>'122,124')  
  
// вызываемый SQL-запрос с параметрами, передаваемыми отдельным блоком  
select \* from fias.1.0.addrobj\_func(settings\_for\_system\_time\_finished=>?, p1=>?, p2=>?)  
params: {"type": "STRING", "value": "122,124"},{"type": "STRING", "value": "001"},{"type": "STRING", "value": "002"}  
  
// SQL-выражение РЗ  
SELECT <атрибуты> FROM <витрина>.<таблица> WHERE id=:p1 AND region=:p2

### 3.3.4 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 [разделе 3.1](#_3.1_SQL-синтаксис).

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.

#### 3.3.4.1 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 [разделе 2.3.1.3](#_2.3.1.3_Выполнение_запроса).

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 для вызова регламентированного 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.

1. 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

#### 3.3.4.2 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 [разделе 2.3.3.1](#_2.3.3.1_Пример_использования).

### 3.3.5 Получение двоичных объектов в результатах запроса

ПОДД СМЭВ предоставляет возможность получения в качестве результата выполнения регламентированных SQL-запросов:

* скалярных параметров (значение которых представляет собой число, строку или дату);
* двоичных объектов (соответствующих файлам, размещенным на Витринах Потребителей данных).

Для получения в качестве ответа двоичных объектов необходимо выполнение следующих условий:

* Витрина Поставщика данных поддерживает тип данных «двоичный объект»;
* в метаданных Витрины Поставщика данных, зарегистрированной в ПОДД СМЭВ, для соответствующих атрибутов должен быть установлен тип данных «двоичный объект» (описание типов данных ПОДД СМЭВ приведено в [разделе 1.5.5](#_1.5_Метаданные_ПОДД)).

Процесс выполнения такого SQL-запроса приведён на рисунке 14, описание процесса приведено далее.

1. ИС Потребителя передаёт SQL-запрос в Агент СМЭВ4.
2. Далее приём, обработка и передача запроса:

Агент Потребителя данных → Ядро ПОДД СМЭВ → Агент Поставщика данных → Витрина Поставщика данных осуществляется аналогично обычному информационному обмену с использованием Регламентированных SQL-запросов (См. [Раздел 1.4.1](#_1.4.1_Обмен_с)).

1. Витрина данных осуществляет обработку запроса и возвращает ответ.

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:

* непосредственно сам двоичный объект;
* уникальная ссылка на получение двоичного объекта с Витрины Поставщика данных (см. шаги [Рисунок - 3.1](#_1d71c45fd73270e1944f8d16b38afcb4)).

Решение о том, какое значение будет передано (объект или ссылка на него), принимается Витриной данных).

1. Далее ответ доставляется до ИС Потребителя аналогично обычному обмену (см. [Раздел 1.4.1](#_1.4.1_Обмен_с)).

Пример возврата двоичного объекта и ссылки на двоичный объект приведен в [разделе 2.3.1.1.3](#_2.3.1.1.3_Ответ_на).

1. Если в результате выполнения запроса передана ссылка, то для получения двоичного объекта ИС Потребителя направляет запрос на получение вложения по ссылке, которую получил в ответе.

Запрос двоичного объекта по ссылке считается отдельным обменом, идентификатор обмена (requestId) запроса по ссылке уникальный (в случае нескольких ссылок - уникальный для каждой из ссылок).

Описание запроса на получение двоичного объекта по ссылке:

* через REST-интерфейс приведено в [разделе 2.3.1.3](#_2.3.1.3_Выполнение_запроса);
* через JDBC-интерфейс приведено далее.

1. Далее приём, обработка и передача запроса Агент Потребителя данных → Ядро ПОДД СМЭВ → Агент Поставщика данных → Витрина Поставщика данных осуществляется аналогично обычному обмену (См. [Раздел 1.4.1](#_1.4.1_Обмен_с)). Для передачи запроса двоичного объекта на Витрину Поставщика используется специальный топик blob.rq.
2. В случае если в теле запроса содержится ссылка на двоичный объект:

* адаптер PODD Витрины данных отправляет запрос в BLOB-адаптер на получение этого файла;
* BLOB-адаптер считывает ссылку на двоичный объект и обращается в Хранилище BLOB-объектов на стороне Ведомства;
* после получения двоичного объекта возвращает его в ПОДД СМЭВ.

Для передачи ответа на запрос двоичного объекта от Витрины Поставщика данных используется специальные топики blob.rs и blob.err.

1. 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.

Пример программного кода для JDBC-драйвера, 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))  
 }  
 }  
}

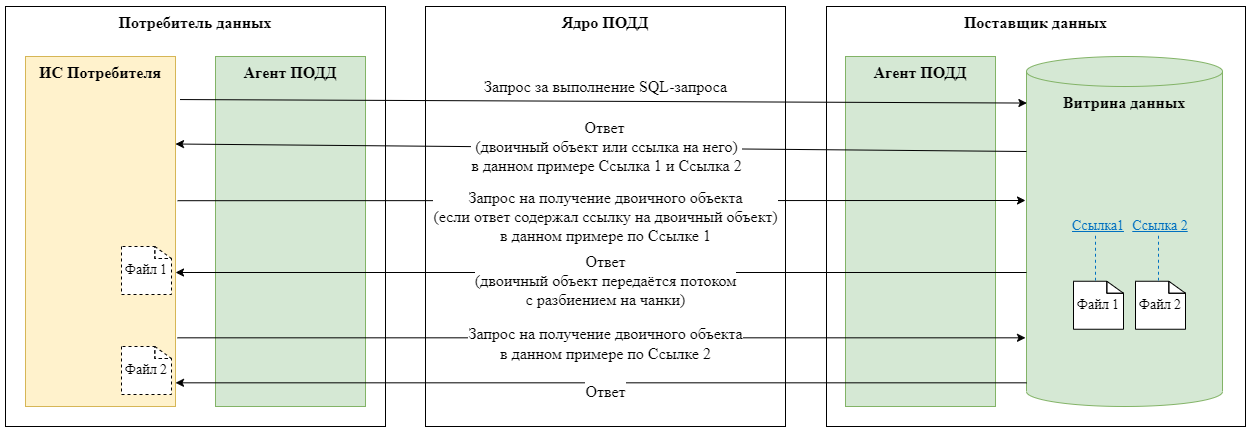


Рисунок - 3.1 Информационный обмен при выполнении запроса с использованием ПОДД СМЭВ

### 3.3.6 Получение печатной формы в результатах запроса

Запросы для получения печатной формы в результатах запроса не содержат SQL-выражения в определении регламентированного SQL-запроса и являются процедурами, выполняемыми на стороне Витрины. Вызов запроса от ИС Потребителя аналогичен вызову обычного регламентированного SQL-запроса.

Приём, обработка и передача запроса по пути Агент Потребителя данных → Ядро ПОДД СМЭВ → Агент Поставщика данных осуществляется аналогично обычному информационному обмену с использованием регламентированных SQL-запросов (См. [Раздел 1.4.1](#_1.4.1_Обмен_с)). Особенностью обработки запросов является добавление Агентом Поставщика данных алиаса сертификата в запрос, согласно настройкам, заданным в соответствии с Руководством администратора ПОДД СМЭВ [[28]](#footnote-28), для последующей подписи печатной формы.

## 3.4 Выполнение запросов к REST-сервису ИС Ответчика

Для выполнения запроса к REST-сервису ИС Ответчика необходимо:

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

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

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

<HTTP-метод> <адрес>:<порт>/<systemMnemonic><basePath><path>

где:

* HTTP-метод – метод из поддерживаемых REST-сервисом;
* <адрес> – IP-адрес Агента Инициатора;
* <порт> – порт для обращения Агента Инициатора к Ядру ПОДД СМЭВ в соответствии с Руководством администратора ПОДД СМЭВ [[29]](#footnote-29);
* systemMnemonic – мнемоника Агента Ответчика, на стороне которого развернут REST-сервис;
* basePath – префикс в URL соответствующего REST-сервиса ИС Ответчика;
* path – путь операции, указанный в спецификации OpenAPI соответствующего REST-сервиса ИС Ответчика.

Пример URI запроса соответствует примеру спецификации OpenAPI, приведенному в [разделе 1.5.5](#_1.5.5_REST-сервисы_ИС):

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

Выполнение запросов осуществляется через REST-интерфейс Агента СМЭВ4 (см. [Раздел 2.3.2](#_2.3.2_REST-интерфейс_Агента)).

## 3.5 Сквозная идентификация запросов

Для диагностики проблем, возникающих в ходе информационных обменов через ПОДД СМЭВ в части отслеживания всей цепочки сообщений, возникающих в ходе информационного обмена, обеспечена сквозная идентификация запросов.

Сквозная идентификация в ПОДД СМЭВ основывается на идентификаторах, приведённых в [таблице 3.3](#_92c98e50d01d5becec3af6e171130627).

На [рисунке - 3.2](#_650792f1011ed5ecf225d533ec926e2f), [рисунке - 3.3](#_c03e94870148657256e37a273468f8fe) и [рисунке - 3.4](#_bd1616cc51f47a97b614f7f6313153d7) приведён порядок возникновения этих идентификаторов.

Таблица 3.3 Виды идентификаторов ПОДД СМЭВ

| № | Вид | Код | Источник идентификатора |
| --- | --- | --- | --- |
| 1 | Клиентский идентификатор  (идентификатор клиента) | clientRequestId | Инициатор обмена |
| 2 | Идентификатор обмена  (идентификатор процесса) | requestId | ПОДД СМЭВ при получении запроса на:   * информационный обмен (например, выполнение регламентированного SQL-запроса) * процесс (например, регистрации подписки) |
| 3 | Идентификатор подзапроса ПОДД СМЭВ – Витрина | subRequestId | ПОДД СМЭВ |
| 4 | Идентификатор подзапроса Адаптер – Prostore | dataRequestId | Адаптер Витрины |

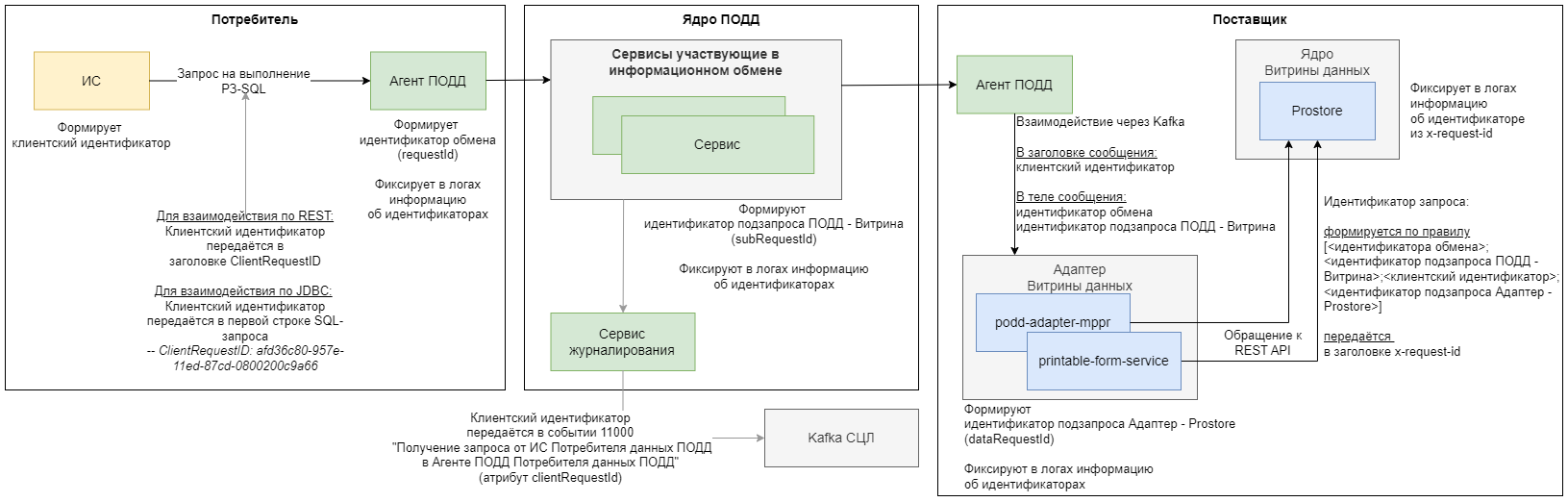


Рисунок - 3.2 Сквозная идентификация запросов для информационного обмена с использованием регламентированных SQL-запросов

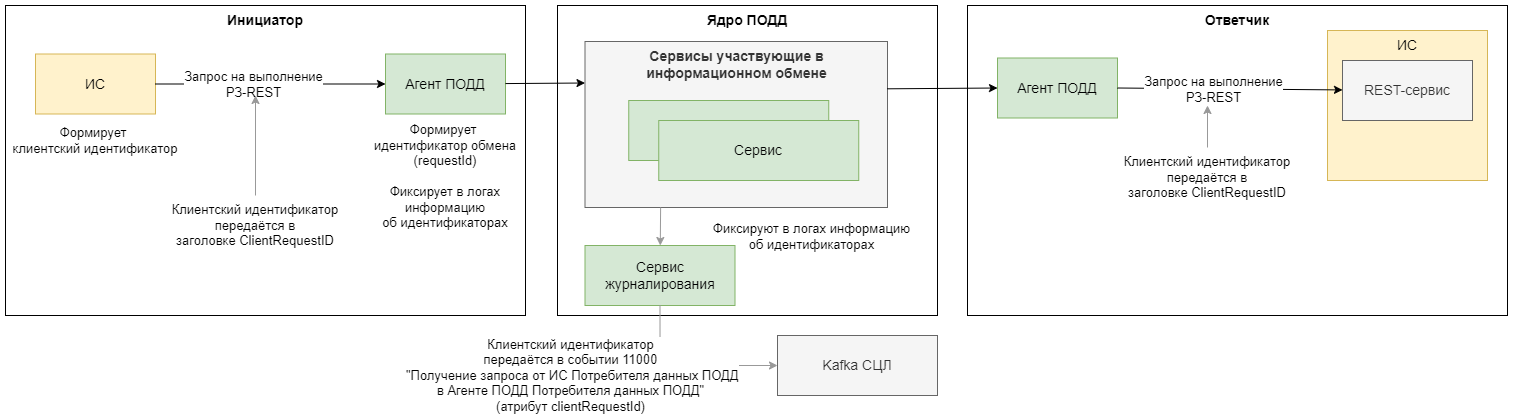


Рисунок - 3.3 Сквозная идентификация запросов для информационного обмена с использованием запросов к REST-сервису ИС Ответчика

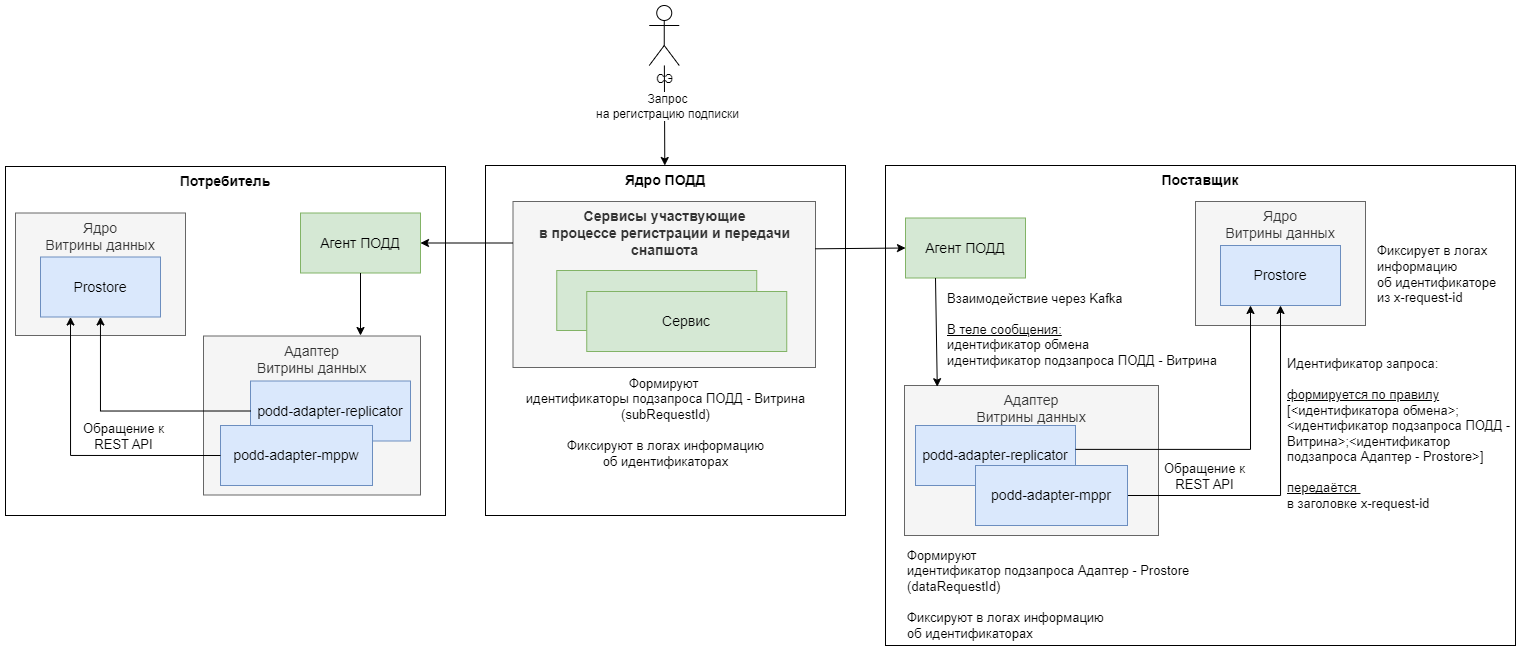


Рисунок - 3.4 Сквозная идентификация запросов для информационного обмена с использованием Рассылок

**Правила использования клиентского идентификатора**

1. Клиентский идентификатор опционален.

*Исключение из правила* – УВ явно добавил обязательное поле ClientRequestID в спецификацию OpenAPI для информационного обмена с использованием регламентированных REST-запросов. В данном случае валидация осуществляется в соответствии со спецификацией.

1. Клиентский идентификатор должен соответствовать стандарту UUID.
2. Ответственность за используемый формат несёт инициатор обмена. ПОДД не осуществляет проверку по формату и версиям UUID.

Исключение из правила – УВ явно добавил поле ClientRequestID в спецификацию OpenAPI для информационного обмена с использованием Регламентированных REST-запросов. В данном случае валидация осуществляется в соответствии со спецификацией.

1. Допустимо использовать один клиентский идентификатор для связывания нескольких запросов. ПОДД СМЭВ не осуществляет проверку на уникальность.
2. Способ передачи клиентского идентификатора приведён в [разделе 2.3](#_2.3_Протокол_взаимодействия).

## 3.6 Лимитирование регламентированных запросов Потребителя к информационным ресурсам Поставщика

### 3.6.1 Лимиты

Для защиты Поставщиков от избыточных запросов Потребителей обеспечена возможность лимитирования запросов к Поставщику данных от конкретного Потребителя данных для информационных обменов с использованием регламентированных запросов.

Лимит может быть установлен на следующие информационные ресурсы Поставщика:

* на конкретный регламентированный SQL-запрос;
* на все регламентированные SQL-запросы к конкретной Витрине;
* на конкретный регламентированный REST-запрос (спецификацию OpenAPI ИС Поставщика);
* общий на все регламентированные SQL и REST-запросы от ИС Потребителя к ИС Поставщика.

Лимит может быть следующих типов:

* на количество запросов;
* на размер запросов;
* на размер ответов.

При создании лимита указывается его значение (количество или размер) и период (у которого два назначения – период контроля соблюдения значения и период блокировки).

### 3.6.2 Условия для добавления лимитов

1. В ПОДД должны быть зарегистрированы ИС Поставщика и ИС Потребителя;
2. Если лимит задаётся на Витрину, в ПОДД СМЭВ должны быть:

* зарегистрирована Витрина;
* Витрина связана с ИС Поставщика;

1. Если лимит задаётся на конкретный регламентированный SQL-запрос, в ПОДД СМЭВ должны быть:

* зарегистрирована Витрина;
* Витрина связана с ИС Поставщика;
* выданы права ИС Потребителя на регламентированный SQL-запрос.

1. Если лимит задаётся на конкретную спецификацию OpenAPI в ПОДД СМЭВ должны быть:

* зарегистрирована спецификация OpenAPI;
* выданы права ИС Инициатора на спецификацию OpenAPI.

1. При выставлении лимитов недопустимо совмещение:

* общего на все регламентированные SQL и REST-запросы ИС Потребителя к ИС Поставщика лимита и лимита на другие ресурсы;
* лимита на Витрину и лимита на регламентированный SQL-запрос к этой Витрине.

1. Лимит на конкретный регламентированный SQL-запрос ограничивает запросы Потребителя по всем версиям этого запроса.
2. Лимит на все регламентированные SQL-запросы к конкретной Витрине ограничивает запросы Потребителя по всем версиям модели этой Витрины.
3. При установке лимита на Витрину и ИС также могут учитываться произвольные SQL-запросы. Реализация исключения таких запросов из процесса лимитирования не предусмотрена, так как права доступа на выполнение произвольных SQL-запросов предоставляются автоматически и только владельцам Витрин на их собственные Витрины.

### 3.6.3 Блокировки

В случае превышения лимитов, заданных Поставщиком, ПОДД СМЭВ блокирует Потребителю доступ к соответствующему информационному ресурсу. Например, если для ИС Потребителя указано два лимита к двум регламентированным SQL-запросам и заблокирован доступ к одному из них, она может продолжить пользоваться другим доступным запросом. При получении запроса от заблокированного Потребителя ПОДД СМЭВ возвращает ответ с соответствующей ошибкой. Блокировка снимается:

* при истечении времени блокировки;
* при удалении соответствующего лимита;
* при редактировании соответствующего лимита (любых атрибутов, кроме наименования);
* при получении запроса на принудительное снятие блокировки.

### 3.6.4 Управление лимитами и блокировками

Управление лимитами и блокировками доступно из ЛК УВ.

### 3.6.5 Алгоритм лимитирования

Лимитирование осуществляется в соответствии со следующим алгоритмом ([Рисунок - 3.5](#_b2eccc078ef492c65ce77633b96f3991)):

1. ПОДД СМЭВ формирует окна наблюдения от 1 января 1970 года 00:00:00.000 в соответствии с указанным значением периода лимита.
2. ПОДД СМЭВ хранит в памяти значение счётичка предыдущего и текущего окна наблюдения. После разблокировки значения счётчиков обнуляются.
3. ПОДД СМЭВ вычисляет счётчик, инкрементируя значение при каждом полученном запросе.
4. ПОДД СМЭВ принимает решение о блокировке на основании имеющихся счётчиков:

* если значение предыдущего окна наблюдения нет, то берётся значение инкрементированного счётчика текущего окна;
* если полученное значение меньше значения лимита, то запрос будет обработан;
* если значение предыдущего окна наблюдения есть, то ПОДД СМЭВ вычисляет коэффициенты наложения окна наблюдения на интервалы по формуле:

K = N + M \* R  
где:  
- N - количество запросов в текущем интервале (с учетом полученного),  
- M - количество запросов на предыдущем интервале,  
- R - коэффициент наложения на предыдущий интервал (0...1).

* если полученное значение меньше значения лимита, то запрос будет обработан.

Пример:

Установлен лимит: 4 запроса в минуту  
С 0 до 0:59 - пришло 5 запросов,  
В 1:30 - пришёл ещё один запрос  
  
результат вычисления:  
- N = 1,  
- M = 5,  
- R = (90 - 60) / 60 = 0.5  
  
K = 1 + 5 \* 0.5 = 3.5 меньше предельного, запрос будет обработан.

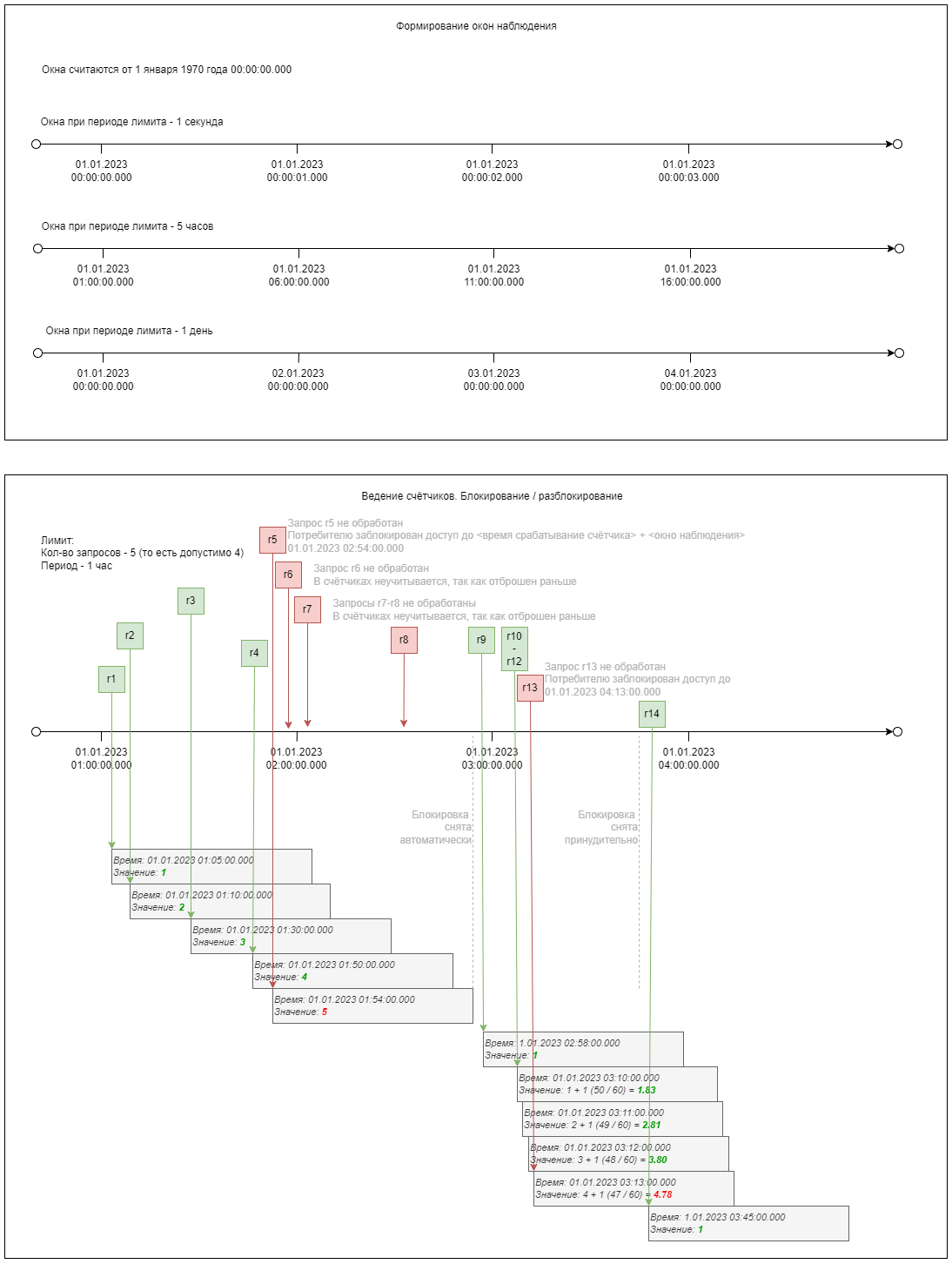


Рисунок - 3.5 Алгоритм лимитрирования запросов

# 4 Термины и сокращения

| Сокращение | Термин | Определение |
| --- | --- | --- |
| Агент СМЭВ4 (Агент) | 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, Регламентированного SQL-запроса) |
|  | 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 | 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, Регламентированного SQL-запроса), 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» |  |
| ЕСКС | Единая система контекстных справок | Доступна по адресу - <https://info.gosuslugi.ru/> |
|  | Запрос данных, SQL-запрос | Произвольный или регламентированный запрос к данным, сформулированный на языке SQL |
|  | Инициатор (запроса) | Участник взаимодействия, который формирует исходный запрос к REST-сервису ИС Ответчика для передачи через ПОДД СМЭВ и может являться источником данных |
| 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 регламентированного SQL-запроса. |
|  | 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. |
|  | 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 |
|  | Ответчик | Участник взаимодействия, на стороне которого развернут REST-сервис ИС Ответчика и который является получателем запроса |
| 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 |
| ПФ | Печатная форма | Документ в формате PDF или XML, сформированный в ответ на поступивший запрос сервисом Витрины на основе предварительно подготовленных pebble-шаблонов, с возможностью добавления к сформированным документам ЭП |
| ПО | Программное обеспечение |  |
|  | Протокол ПОДД СМЭВ | Протокол взаимодействия между Агентом ПОДД и Ядром ПОДД, который представляет собой (в зависимости от вида информационного обмена):   * Бинарный протокол Apache Pulsar с предварительной аутентификацией. * Бинарный протокол RSocket. |
| 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) | Предоставление права Потребителю данных ПОДД СМЭВ на информационный обмен с использованием Регламентированного запроса типа «Рассылка». |
|  | Полная мнемоника регламентированного запроса | Мнемоника, сформированная по правилу <мнемоника витрины>.<мнемоника Регламентированного SQL-запроса>.  Если запрос распределенный, то формируется по правилу podd.<мнемоника Регламентированного SQL-запроса> |
|  | 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 |
|  | 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 запрос | Запрос, определяющий набор данных для обмена между участниками взаимодействия |
| Рассылка | Регламентированный запрос типа «Рассылка» | Регламентированный запрос, содержащий SQL-выражение, выраженное в терминах Модели данных, обеспечивающий автоматическое размещение и актуализацию изменившихся данных из Витрины Поставщика данных в Витрине на стороне Потребителя данных |
| Запрос к REST-сервису ИС Ответчика | Регламентированный запрос типа «Rest-сервис» | Регламентированный запрос, представляющий из себя Rest-сервис в инфраструктуре Ответчика, обеспечивающий предоставление данных другим Участникам взаимодействия с использованием ПОДД СМЭВ в соответствии с загруженной в ПОДД СМЭВ спецификацией OpenAPI |
| Регламентированный SQL-запрос | Регламентированный запрос типа «SQL- Запрос» | Регламентированный запрос, содержащий в себе SQL-выражение, выраженное в терминах Модели данных, загруженное в ПОДД, и зарегистрированное в Ядре ПОДД под символической мнемоникой, используемой ИС Потребителя ПОДД для его выполнения |
|  | 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.digital.gov.ru.ru](https://sc.digital.gov.ru.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 |
| УВ | Участник взаимодействия | Орган или организация, участвующие 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 |
|  | 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 |
|  | Apache Kafka | Распределённый программный брокер сообщений, проект с открытым исходным кодом, разрабатываемый в рамках фонда Apache |
|  | Apache Pulsar | Распределённый программный брокер сообщений, проект с открытым исходным кодом, разрабатываемый в рамках фонда Apache |
| 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 |
| 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 |

# 5 История изменений

Таблица 5.1 История изменений

| Версия | Доработки | Изменения документа |
| --- | --- | --- |
| 2.11.0 | 1. [PODD-3480] - Передача запросов с табличными параметрами в Витрину (обновление) 2. [PODD-4413] - Расширение состава идентифицирующих атрибутов протокола ПОДД | 1. Требования к УВ актуализированы и распределены по разделам соответствующих обменов  * 1.4.1.2, 1.4.2.2, 1.4.3.2 Требования к участникам взаимодействия  1. Добавлена рекомендация по работе с новыми версиями Витрин и РЗ (предупреждение об автоматическом предоставлении доступа к новой версии)  * 1.5.1. Модель данных Витрины Поставщика данных * 1.5.3. Определения Регламентированных запросов  1. Актуализирован перечень заголовков  * 2.3.1. Перечень заголовков сообщений в Apache Kafka  1. Добавлен новый топик для передачи ТП в Витрину с бинарным разбиением на чанки (<мнемоника Витрины>.query.tp.bin)  * 2.3.2. Перечень топиков брокера сообщений Apache Kafka  1. Изменен порядок атрибутов в сообщении с табличными параметрами для Витрины  * 2.3.5.3.1.2. Передача чанков табличных параметров (<мнемоника Витрины>.query.tp / <мнемоника Витрины>.query.tp.bin)  1. Удален атрибут priority из запроса БЛОБ по ссылке  * 2.3.5.3.2.1. Запрос на получение БЛОБ по ссылке от Витрины (<мнемоника Витрины>.blob.rq) |
| 2.12.0 | 1. [PODD-3480] - Реализация Rest для управления метаданными Витрин 2. [PODD-4416] - Управление метаданными РЗ через ЛК УВ Q3 3. [PODD-4735] - Корректировки протокола Агент-Витрина (передача идентифицирующих атрибутов) | 1. Добавлена информация о расширении модели данных Витрины массивом уникальных ключей  * 1.5.1. Модель данных Витрины Поставщика данных  1. Добавлены идентифицирующие атрибуты customerId, customerOgrn, queryMnemonic в структуру сообщений для взаимодействия с Поставщиком   2.3.5.3.1.1. Запрос данных из Витрины (<мнемоника Витрины>.query.rq / <мнемоника Витрины>.procedure.query.rq)  2.3.5.3.1.2. Передача чанков табличных параметров (<мнемоника Витрины>.query.tp / <мнемоника Витрины>.query.tp.bin)   1. Актуализированы ограничения на загружаемые метаданные (на формат и бизнес ограничения)  * 1.5.1. Поддерживаемые форматы * 1.5.2. Модель данных Витрины Поставщика данных * 1.5.4. Определения Регламентированных запросов  1. Скорректирован пример выполнения РЗ (в рамках отказа от \* в SQL РЗ): - 3.2.6. Регламентированные запросы |
| 2.15.0 | 1. [PODD-4927] - Не требуется timestamp перед датой в for\_system\_time 2. [PODD-4928] - Прокидывать единицу времени (day и др) при операциях с датами | 1. Добавлена информация о поддержке FOR SYSTEM\_TIME без TIMESTAMP и вычисления интервала в указанной единице времени, символ \* удален из примеров поддерживаемых SQL-запросов  * 3.1. Поддерживаемые возможности SQL-синтаксиса  1. Актуализация информация о работе с моделью данных Витрины  * 1.5.2. Модель данных Витрины Поставщика данных  1. Добавлено исключение по использованию \* внутри операторов для SQL-выражения РЗ  * 1.5.4. Определения Регламентированных запросов  1. Произвольные SQL-запросы доступны только для владельцев Витрин данных  * 1.5.6.1. Полномочия на выполнение произвольных SQL-запросов |
| 3.1.0 | 1. [PODD-4932] - Передача вложений по API Gateway (запросы к REST-сервису ИС Ответчика) более 5 мб 2. [PODD-4891] - Именованные параметры в Регламентированных SQL-запросах 3. [PODD-4327] - Реализация распределенных подписок | 1. Добавлены требования к УВ для отправки запросов к REST-сервису ИС Ответчика более 5 Мб  * 1.4.3.2.1. Требования к Ответчикам * 1.4.3.2.2. Требования к Инициаторам запросов  1. Обновлена терминология по запросам к REST-сервису ИС Ответчика  * 1.3. Участники информационного обмена с использованием ПОДД СМЭВ * 1.4.3. Обмен с использованием запросов к REST-сервису ИС Ответчика * 1.5. Метаданные ПОДД СМЭВ  1. Добавлено описание механизма прохождения запросов к REST-сервису ИС Ответчика более 5 Мб  * 1.4.3.3. Взаимодействие участников обмена * 1.4.3.4.3. Информационный обмен для механизма с возможностью отправки большого запроса  1. Добавлена информация об именованных параметрах Регламентированных SQL-запросов  * 1.5.4. Определения Регламентированных SQL-запросов * 3.3.3. Примеры вызова и преобразования исходного запроса * 3.3.4. Выполнение запроса с системным параметром  1. Обновлено описание информационного обмена по рассылкам с учётом возможности распределённой подписки, применена согласованная терминология  * 1.4.2. Обмен с использованием Рассылок * 1.5.3. Подписки Потребителей данных ПОДД СМЭВ * 1.5.6.4. Полномочия на Рассылку * 2.3. Протокол взаимодействия Агента СМЭВ4 и Витрины Поставщика данных * 2.5. Протокол взаимодействия Агента СМЭВ4 и Хранилища данных по подписке Потребителя данных |
| 3.2.0 | 1. [PODD-4890] - Доработка конфигурирования Агента ПОДД и схем Витрин (возможность использования общих топиков для нескольких схем Витрин) 2. [PODD-5280] - Вызов РЗ с именованными параметрами для ЦА 3. [PODD-5372] - Ошибка регистрации регламентированного запроса с мнемоникой, содержащей зарезервированное ключевое слово SQL | 1. Добавлена информация о возможности использования общего набора топиков протокола взаимодействия Агент СМЭВ4-Витрина  * 2.3. Протокол взаимодействия Агента СМЭВ4 и Витрины Поставщика данных * 2.5. Протокол взаимодействия Агента СМЭВ4 и Хранилища данных по подписке Потребителя данных  1. Добавлена возможность вызова РЗ через CALL  * 3.3. Выполнение регламентированных SQL-запросов  1. Запрещено использование зарезервированных ключевых слов SQL в мнемонике Витрины и Регламентированного SQL-запроса  * 1.5.1. Поддерживаемые форматы |
| 3.3.0 | 1. [PODD-4885] - Лимиты для запросов к REST-сервису ИС Ответчика | 1. Добавлены возможные коды возврата ПОДД при ошибках выполнения запроса к REST-сервису ИС Ответчика  * 2.4.1.3. Выполнение запросов к REST-сервису ИС Ответчика  1. Добавлена информация о работе и задании лимитов для запросов к REST-сервису ИС Ответчика  * 1.4.3.2.1. Требования к Ответчикам * 1.4.3.4.3. Информационный обмен для механизма с возможностью отправки большого запроса |
| 3.4.0 | 1. [PODD-5435] – Использование диагностического запроса к Витрине данных 2. [PODD-5470] – Обеспечение поддержки времени жизни запроса (таймаут) | 1. Добавлена информация о понятии Диагностический запрос и его использовании  * 3.6. Использование диагностического запроса к Витрине данных * ТЕРМИНЫ И СОКРАЩЕНИЯ  1. Добавлена информация о времени жизни запроса (таймауте) по умолчанию  * 2.4.1.1.1. HTTP-запрос (метод POST) (синхронный режим) * 2.4.1.2.1.1. HTTP-запрос (метод POST) (асинхронный режим) * 2.4.1.2.1.2. HTTP-запрос с табличным параметром (метод POST) (асинхронный режим)  1. Добавлены дополнительные сведения по использованию подписок на Рассылку:  * 1.4.2.1. Общее описание информационного обмена (подписка не может быть поставлена на паузу) * 1.5.3. Подписки Потребителей данных ПОДД СМЭВ на регламентированный запрос типа «Рассылка» (При перерегистрации подписки требуется указывать новый идентификатор) |
| 3.5.0 | 1. [PODD-5459] - Конфигурирование функций агента ПОДД 2. [PODD-5745] - Запрос BLOB по ссылке при взаимодействии ИС Потребителя с Агентом через REST-интерфейс 3. [PODD-5598] - Сквозная идентификация запросов | 1. Добавлена информация о статической и динамической регистрации Витрин:  * 2.3.3. Настройка Агента СМЭВ4 для работы с несколькими Витринами  1. Добавлена информация по запросу BLOB по ссылке через REST-интерфейс:  * 2.4.1. REST-интерфейс Агента СМЭВ4 * 2.4.1.3. Выполнение запроса на получение BLOB по ссылке * 3.3.6. Получение двоичных объектов в результатах запроса  1. Добавлена информация по запросам через REST-интерфейс при подключении к брокеру:  * 2.4.1. REST-интерфейс Агента СМЭВ4  1. Добавлено требование к использованию лимитов (limit), при вызове запросов не являющихся аналитическими:  * 3.3.1. Правила выполнения  1. Добавлены примеры вызова Регламентированных SQL-запросов с параметрами типов TIMESTAMP и DATE:  * 3.3.3. Примеры вызова и преобразования исходного запроса  1. Добавлена информация о сквозной идентификации запросов в ПОДД СМЭВ:  * 3.7. Сквозная идентификация запросов * 2.4. Протокол взаимодействия Агента СМЭВ4 и ИС Потребителя данных (формат передачи клиентского идентификатора) |
| 3.6.0 | 1. [PODD-6320] - [3ЛП][ПОДД] При регистрации РЗ в ЕИП НСУД через REST пропали префиксы таблицы в атрибутах | 1. Добавлены требования к алиасам возвращаемого Потребителю атрибута:  * 1.5.1. Поддерживаемые форматы * 1.5.4. Определения Регламентированных SQL-запросов |
| 3.7.0 | 1. [PODD-5442] – [Проектирование+Реализация] процесс управления подписками в ЛК УВ | 1. Обновлены метаданные подписки и инструменты для управления ими; термин «отмена подписки» заменён на «удаление подписки»; добавлены топики для удаления подписки на потребителе:  * 1.4.2. Обмен с использованием Рассылок; * 1.5.3. Подписки Потребителей данных ПОДД СМЭВ на регламентированный запрос типа «Рассылка»; * 1.5.4. Определения Регламентированных SQL-запросов; * 2.5. Протокол взаимодействия Агента СМЭВ4 и Хранилища данных по подписке Потребителя данных; * Приложение 3 2.5. Удаление подписки.  1. Расширено описание процесса получении двоичных объектов в результатах запросов:  * 3.3.6. Получение двоичных объектов в результатах запроса  1. Добавлен топик scl.signal для Хранилища данных по подписке и добавлены атрибуты для сбытия связанных с подписками:  * 2.5.1. Перечень топиков брокера сообщений Apache Kafka * Приложение 2 - 3.6. Получение событий от Витрины * Приложение 3 - 2.6. Получение событий от Витрины  1. Убрана информация о кодах событий СЦЛ в сообщениях Витрины, в связи с расширением их перечня и неконтролируемостью на стороне ПОДД СМЭВ:  * Приложение 2. - 3.6. Получение событий от Витрины |
| 3.8.0 | 1. [PODD-7151] – Расширение перечня системных параметров 2. [PODD-7039] – Доработка подписания запросов печатных форм 3. [PODD-7161] – Реорганизация обработки запросов к REST-сервису Ответчика 4. [PODD-6388] – Изменены значения таймаутов | 1. Добавлена информация о системных параметрах settings\_for\_system\_time\_started и settings\_for\_system\_time\_finished:  * Раздел 3.3.3  1. Добавлена информация о запросах с получением печатной формы:  * Раздел 1.5.4 * Раздел 3.3.6 * Раздел 4  1. Изменено максимальное значение таймаута, задаваемого в SQL-запросе, корректировка терминологии в части вызова запроса с надстройками и без надстроек:  * Раздел 2.3.1  1. Изменено время хранения результата запроса, вызванного в асинхронном режиме:  * Раздел 2.3.1  1. Корректировка процесса выполнения запросов к REST-сервису Ответчика: валидация осуществляется Ядром  * Раздел 1.4.3.4.1 * Раздел 1.4.3.4.2  1. Оптимизация документа. Удалены разделы с описанием технических деталей:  * 2.1. Протокол ПОДД СМЭВ * 2.3.1 Перечень заголовков сообщений в Apache Kafka * 2.3.4 Последовательность исполнения SQL-запроса в контуре Поставщика данных * 2.5.2 Структуры сообщений для взаимодействия с Потребителем данных * 3.3.2 Преобразование исходного запроса * 3.5. Формирование и передача статистики атрибутов Витрины данных Агенту СМЭВ4 * Приложение 2 Структуры сообщений для взаимодействия с Поставщиком * Приложение 3 Структуры сообщений для взаимодействия с Потребителем данных  1. Оптимизация документа. Подготовлено описание без технических деталей:  * Раздел 1.4.1.1 * Раздел 4 |
| 3.9.1 | 1. [PODD-5671] – Добавлена возможность лимитирования регламентированных запросов Потребителя к информационным ресурсам Поставщика данных 2. [PODD-6670] – Удалён механизм по управлению подписками через ВС | 1. Добавлена информация о лимитировании запросов в СМЭВ4:  * Раздел 3.6  1. Убрана информация об управлении подписками через ВС:  * Раздел 1.5.3  1. Доработка документа. Скорректировано описание типов атрибутов и параметров:  * Раздел 1.5.1  1. Доработка документа. Указан заголовок с версией протокола REST-интерфейса Агента ПОДД для SQL-запросов:  * Раздел 2.3.1  1. Удален раздел «Приложение 1 Пример проекта для реализации взаимодействия с Агентом СМЭВ» 2. Изменения в терминах:   «Агент ПОДД» заменен на «Агент СМЭВ4» |

1. Документ размещен на портале ЕСКС, <https://info.gosuslugi.ru> [↑](#footnote-ref-1)
2. Перечисленные руководства и регламенты размещены на портале ЕСКС – <https://info.gosuslugi.ru/> [↑](#footnote-ref-2)
3. Перечисленные руководства и регламенты размещены на портале ЕСКС – <https://info.gosuslugi.ru/> [↑](#footnote-ref-3)
4. Документ размещен на портале ЕСКС – <https://info.gosuslugi.ru/> [↑](#footnote-ref-4)
5. Регламентированный запрос типа «Рассылка» и права доступа на него регистрируются аналогично регламентированному SQL-запросу. [↑](#footnote-ref-5)
6. Перечисленные документы размещены на портале ЕСКС – <https://info.gosuslugi.ru/> [↑](#footnote-ref-6)
7. Перечисленные документы размещены на портале ЕСКС – <https://info.gosuslugi.ru/> [↑](#footnote-ref-7)
8. ПО «Витрина данных» поддерживает самостоятельный запрос дельты только для распределённых подписок. [↑](#footnote-ref-8)
9. Перечисленные документы размещены на портале ЕСКС – <https://info.gosuslugi.ru/> [↑](#footnote-ref-9)
10. Перечисленные руководства и регламенты размещены на портале ЕСКС – <https://info.gosuslugi.ru/> [↑](#footnote-ref-10)
11. Подробная информация расположена на портале <https://calcite.apache.org/docs/reference.html#keywords> [↑](#footnote-ref-11)
12. Документ размещен на портале ЕСКС <https://info.gosuslugi.ru> [↑](#footnote-ref-12)
13. Документ размещена на портале ЕСКС <https://info.gosuslugi.ru> [↑](#footnote-ref-13)
14. Документ размещен на портале ЕСКС – <https://info.gosuslugi.ru/> [↑](#footnote-ref-14)
15. Документ размещен на портале ЕСКС – <https://info.gosuslugi.ru/> [↑](#footnote-ref-15)
16. Документ размещен на портале ЕСКС, <https://info.gosuslugi.ru> [↑](#footnote-ref-16)
17. Указанные документы размещены на портале ЕСКС, <https://info.gosuslugi.ru> [↑](#footnote-ref-17)
18. Документ размещен на портале ЕСКС – <https://info.gosuslugi.ru/> [↑](#footnote-ref-18)
19. Документ размещен на портале ЕСКС – <https://info.gosuslugi.ru/> [↑](#footnote-ref-19)
20. Далее в [Разделе 2.3](#_2.3_Протокол_взаимодействия) префикс <префикс> в названии топиков может быть опущен. [↑](#footnote-ref-20)
21. Описание приведено в Руководстве администратора Агента ПОДД СМЭВ, размещенном на портале ЕСКС – <https://info.gosuslugi.ru/> [↑](#footnote-ref-21)
22. Данная информация используется Агентом СМЭВ4 при работе и обновляется на основе актуальных данных от Ядра при каждом перезапуске. [↑](#footnote-ref-22)
23. Документ размещен на портале ЕСКС – <https://info.gosuslugi.ru/> [↑](#footnote-ref-23)
24. Документ размещен на портале ЕСКС – <https://info.gosuslugi.ru/> [↑](#footnote-ref-24)
25. Документ размещен на портале ЕСКС – <https://info.gosuslugi.ru/> [↑](#footnote-ref-25)
26. Документ размещен на портале ЕСКС – <https://info.gosuslugi.ru/> [↑](#footnote-ref-26)
27. Документ размещен на портале ЕСКС – <https://info.gosuslugi.ru/> [↑](#footnote-ref-27)
28. Документ размещен на портале ЕСКС – <https://info.gosuslugi.ru/> [↑](#footnote-ref-28)
29. Документ размещен на портале ЕСКС – <https://info.gosuslugi.ru/> [↑](#footnote-ref-29)