

ДЕПАРТАМЕНТ ЗДРАВООХРАНЕНИЯ
ХАНТЫ-МАНСИЙСКОГО АВТОНОМНОГО ОКРУГА – ЮГРЫ

**Центральный узел регионального сегмента
единой государственной информационной системы в сфере здравоохра-
нения Ханты-Мансийского автономного округа – Югры**

Компонент «Портал пациента ЦУ РС ЕГИСЗ»

**РЕГЛАМЕНТ ИНФОРМАЦИОННОГО ВЗАИМОДЕЙСТВИЯ
«Портал пациента ЦУ РС ЕГИСЗ
(с изменениями в части взаимодействия с подсистемой «Регис-
тратура»))»**

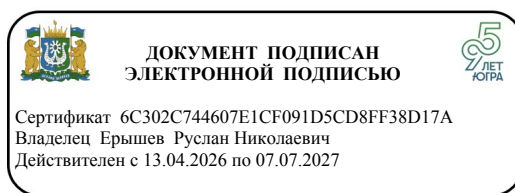
ОПИСАНИЕ ИНТЕГРАЦИОННЫХ ПРОФИЛЕЙ

ЛИСТ УТВЕРЖДЕНИЯ
Версия – 2.8
Версия релиза API ФЭР Шлюз 26.05

15.07.2026

Утверждаю:

Начальник управления
развития системы здра-
воохранения
Департамента здраво-
охранения
Ханты-Мансийского ав-
тономного
округа – Югры



Р.Н. Ерышев

ЛИСТ ИЗМЕНЕНИЙ

Дата	Версия	Описание изменений	Автор
07.02.2023	1.0		Гиззатуллина Г.С.
13.03.2023	1.1	1. Добавлен <i>раздел 7</i>. Методы ответного сервиса 2. Добавлен <i>раздел 8</i>. Адреса сервисов интеграционного шлюза 3. <i>Приложение 1</i> – добавлены изменения в следующие таблицы: • в <i>таблице 2</i> depart_id заменено на mo_id; - SNILS – snils; - lastName – last_name; - firstName – first_name; - middleName – middle_name; • в <i>таблице 4</i>: - visitTime – visit_time • в <i>таблице 6</i>: - depart_id на mo_id; • в <i>таблице 8</i>: - to_depart_id - to_mo_id; - from_depart_id - from_mo_id; - Reason – reason; - Resource_id - resource_id; - Profile_code - profile_code; - Profile_name - profile_name. 4. В <i>Приложении 1</i> добавлены комментарии, обязательность полей. 5. Удален <i>раздел 5.11</i> Метод Observation, таблица 11 – Observation. 6. В <i>разделах 5.1 – 5.10</i> добавлены изменения в методы удаления данных. 7. В <i>разделе 6.1</i>. внесены изменения в наименование параметров: bookId - book_id; slotId - slot_id; caseNumber - case_number; prelimi-	Гиззатуллина Г.С.

		naryReservation - preliminary_reservation; mobile-Phone - mobile_phone; bookExtId - book_ext_id; visitTime - visit_time; serviceId - service_id; organizationId - organization_id; areaId - area_id; queueNumber - queue_number. В разделе 6.2 внесены изменения в наименования параметрpf bookExtId - book_ext_id.	
17.07.2023	1.2	1. Добавлен п. 3.1.4 Требования к наполнению данными. 2. В п. 5 добавлены методы получения и удаления данных. 3. В п. 6.1., в описании параметров в запросе: - удалено поле book_id; - добавлен поля данных о пользователе из РРП. В п. 6.1., в описании параметров в ответе: - добавлено описание поля book_ext_id; - добавлено обязательность поля duration. 4. В п. 6.2 исправлен пример успешного ответа, добавлен пример ответа с бизнес ошибкой. 5. В п. 6.2.1 исправлен код ошибки с 19 на 1. 6. Удалены п. 5.5 Описание подсистемы Patient и п. 5.6 Описание подсистемы Attachment 7. В Приложение 1 добавлены изменения в следующие таблицы: - изменено описание поля update_ts во всех блоках данных; - в таб. 1 МО добавлено поле branch_oid – обязательное; - в таб. 2 Resource изменена обязательность поля enabled; - в таб. 4 Slot изменена обязательность поля visit_time; - в таб. 4 Slot изменена обязательность поля duration; - удалены таблица 5 и таблица 6. 8. Добавлено условие проверки при выгрузке бирки в п 5.7.1.	Гиззатуллина Г.С.

28.08.2023	1.3	1. Добавлены разделы: 6.1.1 Оптимизация процесса бронирования 6.2.1 Оптимизация процесса отмены бронирования 2. Приложение 1 – Добавлен формат для GUID. 3. Приложение 1, табл. 1, branch_oid – добавлены типы подразделений. 4. Приложение 1, табл. 2, post_id – добавлены коды должностей. 5. Приложение 1, табл. 2, post_name – атрибут удален. 6. Раздел 6.2.2 Добавлен код ошибки 503 7. Раздел 5.2.3 Добавлен метод получения данных ресурсов по МО 8. Раздел 5.4.3 Добавлен метод получения данных слотов по ресурсу 9. Раздел 5.7.3 Добавлен метод получения данных бирок по идентификатору слота 10. Раздел 6.2 В описание параметров в запросе добавлено поле united_guids. 11. Приложение 1, табл. 4, mo_id – добавлен атрибут.	Гиззатуллина Г.С.
07.12.23	1.4	1. Приложение 1, табл. 2, добавлен код должности 120 – врач-фтизиатр участковый 2. Добавлен раздел 7.5 Проверка доступности вебхуков. 3. Приложение 2. Изменен форма ответа 4. Обновлен адрес тестового контура. 5. В Приложение 1 добавлены изменения в следующие таблицы: - в таб. 4 Slot удалено поле mo_id.	Гиззатуллина Г.С.
27.03.24	1.5	1. Раздел 5 Добавлена таблица с описанием обязательных атрибутов для заголовка при использовании gzip. 2. Раздел 6.2.2 Добавлен коды ошибок: 405 - Обновление информации заблокировано	Гиззатуллина Г.С.

		422 – Дублирование данных 3. Раздел 5.4 Добавлены проверки при отправке слотов 4. Раздел 5.1 Добавлены проверки при отправке подразделений 5. Добавлен раздел 3.1.5 Требование к удалению данных	
24.05.24	1.6	Изменения для записи по направлению: 1. Приложение 1: В таб. 8 – Referral изменена обязательность полей: speciality_id, to_profile_code, to_post_name, to_resource_id, to_service_id. Удалены атрибуты: to_profile_name, to_service_name. Изменено описание поля type. В таб. 10 – Profilecode_resource удален атрибут profile_name. В таб. 3 – Service удалены атрибуты name, изменена обязательность поля code. 2. Раздел 5.1.3 добавлен метод получения данных о подразделениях МО по OID МО. 3. Добавлен Раздел 7 – Запись по направлению	Гиззатуллина Г.С.
06.06.24	1.7	1. В приложение 1 добавлены изменения в следующие таблицы: - в таб. МО добавлено обязательное поле tag_age	Гиззатуллина Г.С.
23.07.24	1.8	1. Добавлены разделы: 7.5 Описание реализации услуги на ЕПГУ 7.6 Требования к услуге 2. Внесены изменения в таб. 8 – Referral.	Гиззатуллина Г.С.
15.08.24	1.9	1. В приложение 1 добавлены изменения в следующие таблицы: - в таб. Slot добавлено значение young для поля tag_age - в таб. Unaccessable_period добавлены новые значения ASSIGNMENT, OTHER, удалено значение FIRED.	Гиззатуллина Г.С.
01.12.2024	2.0	1. Приложение 1 – добавлены изменения в следующие таблицы: - в таб. 12 добавлено поле source_record; - в таб. 13 удалены поля cancel_reason, status, duty_doctor; - в таб. 4 добавлены новые типы слотов.	Гиззатуллина Г.С.
21.02.25	2.1	1. Приложение 3 – добавлено описание брокера 2. Раздел 6:	Гиззатуллина Г.С.

		- добавлены разделы 6.3 и 6.4 методы бронирования/отмены через Контакт-центр	
23.06.25	2.1.1	1. Исправлена ошибка в описании поля <code>symp_emis</code> в вызове на дом	Елисеев Д.С.
11.09.2025	2.1.2	1. Приложение 3 - отредактировано описание взаимодействия с брокером сообщений Kafka 2. Приложение 1, табл. 1-13 - Добавлено описание атрибутов в топиках Kafka	Дунюшкин Г.К.
28.10.25	2.2	1. В п.5.8.4 добавлено описание метода получения данных о направлении по номеру направления 2. В п.5.8.5 добавлено описание метода получения данных о направлении по id пациента	Дунюшкин Г.К.
14.11.25	2.3	1. В п.6.3 добавлено описание процесса авторизации при обращении к адаптеру бронирования	Дунюшкин Г.К.
05.02.26	2.4	1. Актуализированы все эндпоинты в разделе 5 2. Актуализированы все поля сущностей в приложении 1 3. Добавлены эндпоинты для подсистем Locations и Resource Location (п. 5.14 и п. 5.15) 4. Добавлены описания сущностей подсистем Locations и Resource Location в Приложении 1 5. Добавлены диаграммы последовательностей работы систем в разделе 6, для различных сценариев. 6. Добавлен User case (Приложение 3 раздел 3) по взаимодействию с сообщениями в Kafka	Дунюшкин Г.К.
25.02.26	2.5	1. Добавлено описание подсистемы Observation	Дунюшкин Г.К.
23.03.26	2.6	1. Изменен ip адрес промышленного контура веб-сервиса по приему данных от МО	Дунюшкин Г.К.
20.04.26	2.7	1. В п. 5.2.5 добавлены query-параметры: <code>ergu-list</code> , <code>tag-age</code> . Обновлен пример запроса метода с учетом новых параметров. 2. В п. 5.4 добавлен query-параметр <code>tag-age</code> . Обновлен пример запроса метода с учетом нового параметра. 3. В Приложении 1, Таблица 13 (Home Visit), для поля <code>type_call</code> добавлено новое допустимое значение 4. В Приложении 1, Таблица 15 (Resource Location), добавлено поле <code>is_primary</code> (boolean)	Дунюшкин Г.К.
03.07.2026	2.8	1. В п. 5.16.3 добавлен метод получения данных подсистемы Observation. 2. В Приложении 1, Таблица 2 – Resource уточнена условная обязательность атрибутов для ре-	Дунюшкин Г.К.

		сурсов типа MACHINE: machine_name; добавлен атрибут serial_number. 3. В Приложении 1, Таблица 4 – Slot добавлен атрибут mo_quota для передачи квот головных МО по котируемым слотам. 4. В Приложении 1, Таблица 7 – Book добавлены атрибуты origin_source и last_change_source для учета источника создания и источника последнего изменения записи. 5. В Приложении 1, Таблица 8 – Referral актуализированы обязательность и правила заполнения полей направления: speciality_id, type, to_mo_id, to_profile_code, to_service_name, to_post_name, to_service_id, disease_code; расширен перечень допустимых значений type. 6. В Приложении 4 добавлены правила передачи origin_source и last_change_source, включая сопоставление booking_source с tag_source и last_change_source, а также ограничение tag_source только регламентными значениями.	
--	--	---	--

СОДЕРЖАНИЕ

ЛИСТ ИЗМЕНЕНИЙ	2
ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ	9
АННОТАЦИЯ	11
1 НАЗНАЧЕНИЕ ДОКУМЕНТА	12
2 СХЕМЫ ВЗАИМОДЕЙСТВИЯ РМИС И ЕПГУ	13
2.1 Схемы взаимодействия РМИС и ЕПГУ	13
2.2 Схема взаимодействия компонентов шлюза ФЭР 2.0 и компонентов ПО Витрина данных с сохранением записей в БД и взаимодействием с ЕМИС	14
3 ПОСЛЕДОВАТЕЛЬНОСТЬ ВЗАИМОДЕЙСТВИЯ	15
3.1 Описание сценария «Предоставление сведений в Витрину данных»	15
3.2 Описание сценария «Запись на прием к врачу»	18
3.3 Описание сценария «Отмена записи на прием к врачу»	19
4 СЕРВИС АВТОРИЗАЦИИ	22
4.1 Авторизация к топикам Kafka	23
5 СЕРВИС ВЗАИМОДЕЙСТВИЯ С ПРОГРАММНЫМ ОБЕСПЕЧЕНИЕМ «ВИТРИНА ДАННЫХ»	24
5.1 Описание подсистемы MO	25
5.2 Описание подсистемы Resource	27
5.3 Описание подсистемы Service	29
5.4 Описание подсистемы Slot	31
5.5 Описание подсистемы Patient	35
5.6 Описание подсистемы Attachment	35
5.7 Описание подсистемы Book	35
5.8 Описание подсистемы Referral	37
5.9 Описание подсистемы Unaccessible_period	38
5.10 Описание подсистемы Profilecode_resource	40
5.11 Описание подсистемы Waiting_list	41
5.12 Описание подсистемы Waiting list entity	42
5.13 Описание подсистемы Home Visit	42
5.14 Описание подсистемы Locations	43
5.15 Описание подсистемы Resource Location	45
5.16 Описание подсистемы Observation	46
6 СЕРВИС БРОНИРОВАНИЯ И ОТМЕНЫ ЗАПИСИ НА ПРИЕМ	47
6.1 Метод бронирования записи	47
6.2 Отмена бронирования выбранного слота	49
6.3 Коды ошибок	51
6.4 Авторизация при обращении к адаптеру бронирования	56
7 ЗАПИСЬ НА ПРИЕМ ПО НАПРАВЛЕНИЮ	57
7.1 Краткое описание	57
7.2 Клиентский путь «Запись на прием по направлению в медицинскую организа- цию»	58
7.3 Сценарий «Запись на прием по направлению в медицинскую организацию»	58
7.4 Сценарий «Отмена записи на прием по направлению в медицинскую организа- цию»	61
7.5 Описание реализации услуги на ЕПГУ	62
7.6 Требования к услуге	65
8 ОТВЕТНЫЙ СЕРВИС	68
8.1 Регистрация вебхука	68
8.2 Получение вебхука	69
8.3 Обновление вебхука	69

8.4	Удаление вебхука	70
8.5	Проверка доступности вебхука	70
9	АДРЕСА СЕРВИСОВ ИНТЕГРАЦИОННОГО ШЛЮЗА.....	71
	ПРИЛОЖЕНИЕ 1. СТРУКТУРА СВЕДЕНИЙ СЕРВИСА «ВИТРИНА ДАННЫХ»..	72
	Таблица 1 – МО - структурные подразделения.....	72
	Таблица 2 – Resource - ресурсы МО.....	77
	Таблица 3 – Service – услуги.....	81
	Таблица 4 – Slot - слоты.....	82
	Таблица 7 – Book - записи на бирку пациентов.....	86
	Таблица 8 – Referral - направления пациентов.....	89
	Таблица 9 – Unaccessible_period - периоды недоступности ресурсов.....	94
	Таблица 10 – Profilecode_resource	95
	Таблица 11 – Waiting_list - Лист ожидания.....	97
	Таблица 12 – Waiting_list_entity - Запись в листе ожидания.....	98
	Таблица 13 – Home Visit – Вызовы врачей на дом.....	100
	Таблица 14 – Locations - Участки.....	108
	Таблица 15 – Resource Location – Ресурсы закрепленные за участками.....	109
	Таблица 16 – Observation – Записи по карте на диспансерное наблюдение.....	111
	ПРИЛОЖЕНИЕ 2. СТРУКТУРА ОТВЕТА СЕРВИСА ВЗАИМОДЕЙСТВИЯ С ПО	
	«ВИТРИНА ДАННЫХ».....	113
	ПРИЛОЖЕНИЕ 3. ВЗАИМОДЕЙСТВИЕ С БРОКЕРОМ СООБЩЕНИЙ КАФКА..	114
	ПРИЛОЖЕНИЕ 4. НАСТРОЙКА ПЕРЕДАЧИ ПОЛЕЙ ORIGIN_SOURCE И	
	LAST_CHANGE_SOURCE ДЛЯ ОРГАНИЗАЦИИ УЧЕТА БРОНИРОВАНИЯ..	121

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

Термин/сокращение	Определение
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
Representational State Transfer (REST)	Архитектурный стиль взаимодействия компонентов распределённого приложения в сети
SOAP	Simple Object Access Protocol – простой протокол доступа к объектам
WSDL	Web Services Description Language – язык описания
XSD	XML Schema definition – язык описания структуры XML-документа
Агент (ПОДД СМЭВ)	Типовое программное обеспечение, устанавливаемое в контуре ИС УВ и обеспечивающее сопряжение Витрин данных ИС УВ и ИС УВ с ПОДД СМЭВ
БД	База данных
Витрина данных/Витрина	Комплекс программных и технических средств в составе информационно-телекоммуникационной инфраструктуры Участника взаимодействия, обеспечивающий хранение и предоставление данных другим Участникам взаимодействия с использованием ПОДД СМЭВ
ЕПГУ	Единый портал государственных и муниципальных услуг (функций)
ЕСИА	Единая система идентификации и аутентификации
Запись на прием к врачу	Услуга «Запись на прием к врачу», предоставляемая гражданам посредством информационной системы пользователя в соответствии с пунктом 19 Приложения №2 к Положению о единой государственной информационной системе в сфере здравоохранения, утвержденному постановлением Правительства РФ от 5 мая 2018 № 555 «О единой государственной информационной системе в сфере здравоохранения»
ЗСПД	Защищенная сеть передачи данных
ИС	Информационная система
МО	Медицинская организация
Подсистема обеспечения доступа к данным СМЭВ (ПОДД СМЭВ)	Часть транспортной подсистемы СМЭВ, обеспечивающая доступ к данным, размещённым на Витринах данных или в ИС поставщика данных
Пользователь	Гражданин, обратившийся за получением Услуги к ИС пользователя

Термин/сокращение	Определение
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
Регламентированный запрос	SQL-запрос, выраженный в терминах Модели данных, загруженной в ПОДД, и зарегистрированный в Ядре ПОДД СМЭВ под символической мнемоникой, используемой ИС Потребителя ПОДД СМЭВ для выполнения регламентированного запроса. Может иметь параметры, значения которых задаются Потребителем данных ПОДД СМЭВ при выполнении регламентированного запроса
СНИЛС	Страховой номер индивидуального лицевого счёта
СМЭВ	Система межведомственного электронного взаимодействия
СП МО	Структурное подразделение медицинской организации
СТП	Служба технической поддержки
СЦ	Ситуационный центр
ТФОМС	Территориальный фонд обязательного медицинского страхования
УЗ	Учетная запись
Услуга	Запись на прием к врачу, предоставляемая в электронной форме посредством информационной системы пользователя в соответствии с пунктом 19 Приложения №2 к Положению о единой государственной информационной системе в сфере здравоохранения, утвержденному постановлением Правительства РФ от 5 мая 2018 № 555 «О единой государственной информационной системе в сфере здравоохранения»
Участник взаимодействия (УВ)	Орган или организация, участвующая в информационном обмене через СМЭВ
ФНСИ	Федеральный реестр нормативно-справочной информации в сфере здравоохранения
Ядро ПОДД СМЭВ	Часть ПОДД СМЭВ, предназначенная для проверки и маршрутизации запросов между участниками взаимодействия

АННОТАЦИЯ

Настоящий документ определяет регламент информационного взаимодействия компонента «Портал пациента» центрального узла (ЦУ) единой государственной информационной системы в сфере здравоохранения (ЕГИСЗ) с подсистемой «Регистратура» медицинских организаций посредством интеграционного взаимодействия с региональным сегментом ЕГИСЗ.

Документ описывает формат, порядок и регламент обмена сообщениями между взаимодействующими компонентами с указанием формата представляемых данных и условий взаимодействия.

1 Назначение документа

Настоящий документ предназначен для разработчиков медицинских информационных систем (далее МИС) и единых медицинских информационно-аналитических систем (далее ЕМИС), интегрируемых с компонентом «Портал пациента» централизованной управляющей системы (далее ЦУ РС ЕГИСЗ) и описывает методологию разработки информационного взаимодействия с подсистемой «Регистратура».

2 Схемы взаимодействия РМИС и ЕПГУ

2.1 Схемы взаимодействия РМИС и ЕПГУ

УВ являются – ЕПГУ и РМИС. Общая схема взаимодействия представлена на рисунке Рисунок 1.

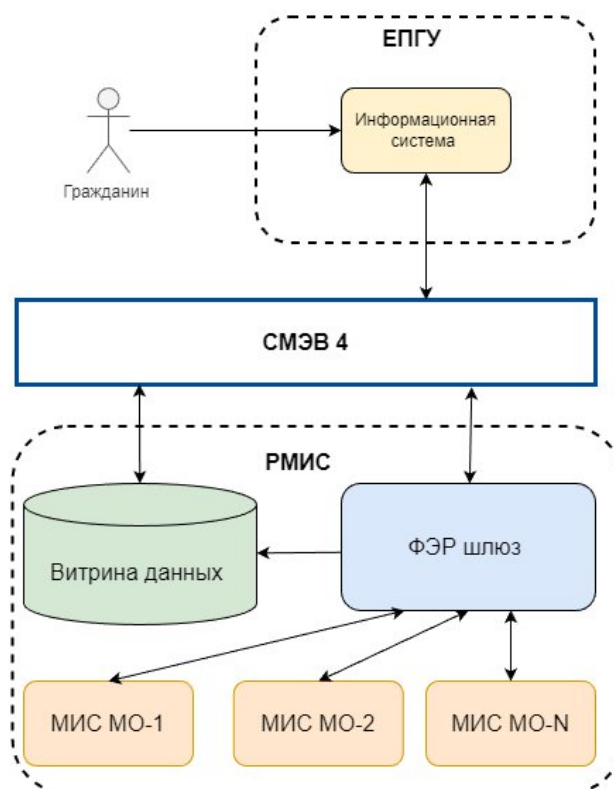


Рисунок 1 – Схема взаимодействия

Потребителем данных выступает ЕПГУ. На его ресурсах развернуты:

- ИС.

Для взаимодействия с СМЭВ, ИС включает в себя REST-сервис, обеспечивающий:

- передачу запроса расписания слотов;
- приём ответа на запрос расписания слотов;
- передачу запроса на бронирование слота;
- передача ответа на запрос бронирования.

В качестве поставщика данных выступает РМИС, на ресурсах которой должны быть развёрнуты:

- Витрина данных «Запись к врачу».

– ИС.

Запрос расписания слотов обеспечивается взаимодействием СМЭВ с Витриной данных «Запись к врачу».

Запрос бронирования слота обеспечивается взаимодействием СМЭВ с ИС через её REST-сервис и ИС с Витриной данных «Запись к врачу».

2.2 Схема взаимодействия компонентов шлюза ФЭР 2.0 и компонентов ПО Витрина данных с сохранением записей в БД и взаимодействием с ЕМИС

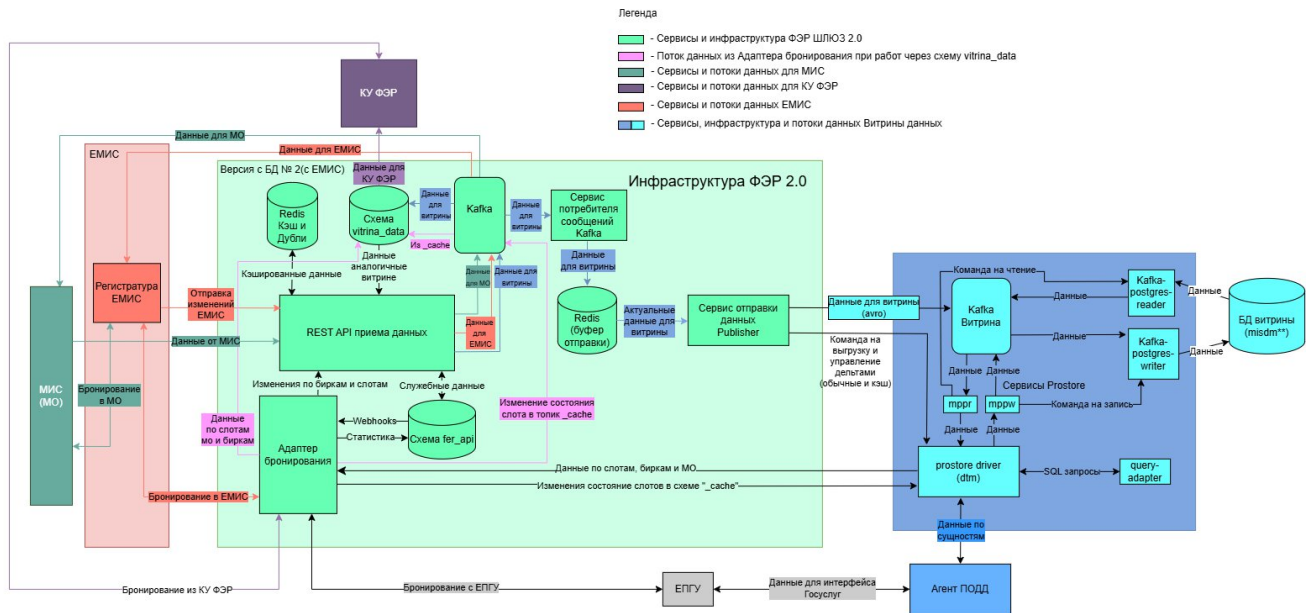


Рисунок 2 – Схема взаимодействия сервисов

3 Последовательность взаимодействия

С помощью компонента «Витрина данных «Запись к врачу» обеспечивается выполнение следующих сценариев:

- Идентификация пользователя и предоставление сведений о пользователе: прикрепление к поликлинике в регионе, прикрепление по диспансерному учету, наличие активных направлений и записей на прием к врачу/процедуры.
- Предоставление сведений о медицинских организациях, медицинских работниках и свободных слотах для записи на прием к врачу, по направлению, к лечащему врачу по диспансерному учету.
- Бронирование слота и запись на прием к врачу, по направлению, к лечащему врачу по диспансерному учету.
- Изменение статуса при отмене пользователем записи к врачу, по направлению, к лечащему врачу по диспансерному учету.
- Предоставление сведений для отображения пользователю текущих записи, сделанных через различные источники записи.

3.1 Описание сценария «Предоставление сведений в Витрину данных»

В данном разделе описывается сценарий «Предоставление сведений в Витрину данных».

3.1.1 Участники

ИС:

- Компонент «Витрина данных «Запись к врачу».
- ФЭР шлюз.
- МО.
- ЕМИС

3.1.2 Критерий успешности выполнения

Данные загружены в ПО «Витрина данных».

Информационный обмен осуществляется следующим образом (диаграмма последовательности приведена на рисунке 3:

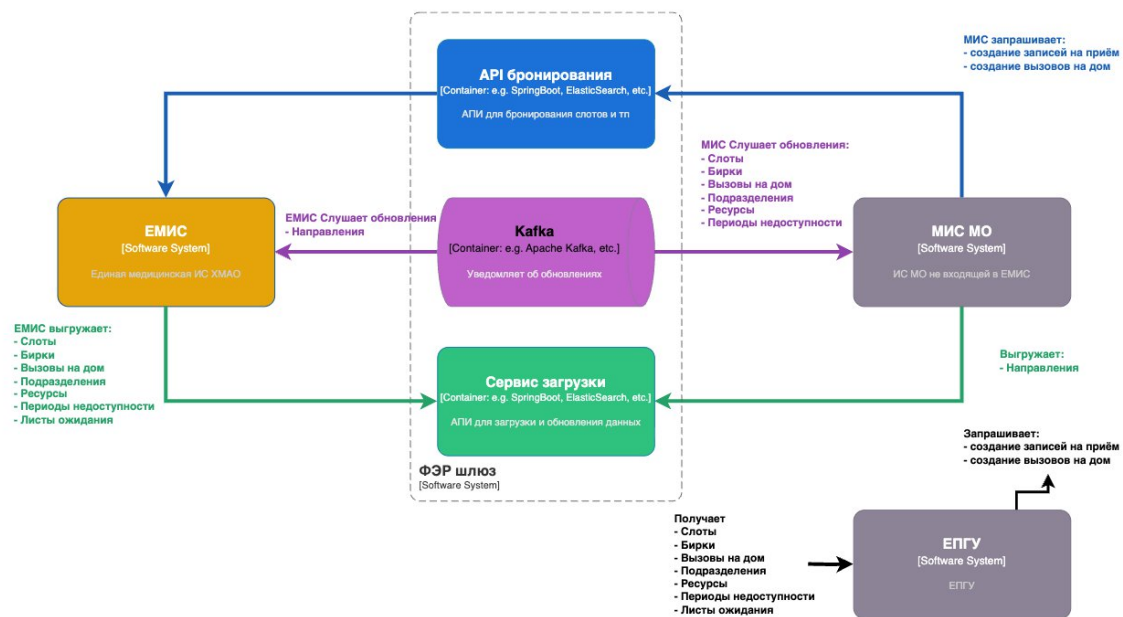


Рисунок 3 – Схема взаимодействия ЕМИС и ФЭР Шлюз

1. ЕМИС отправляет запрос на добавление данных.
2. ФЭР шлюз принимает запрос ЕМИС.
3. ФЭР шлюз проводит ФЛК проверку и отправляет ответ в ЕМИС.
4. ФЭР шлюз загружает данные в ПО «Витрина данных».

3.1.3 Порядок загрузки данных

Для сохранения целостности данных, рекомендуется соблюдать порядок загрузки данных в Витрину.

Оптимальный порядок выгрузки данных представлен на рисунке 4.

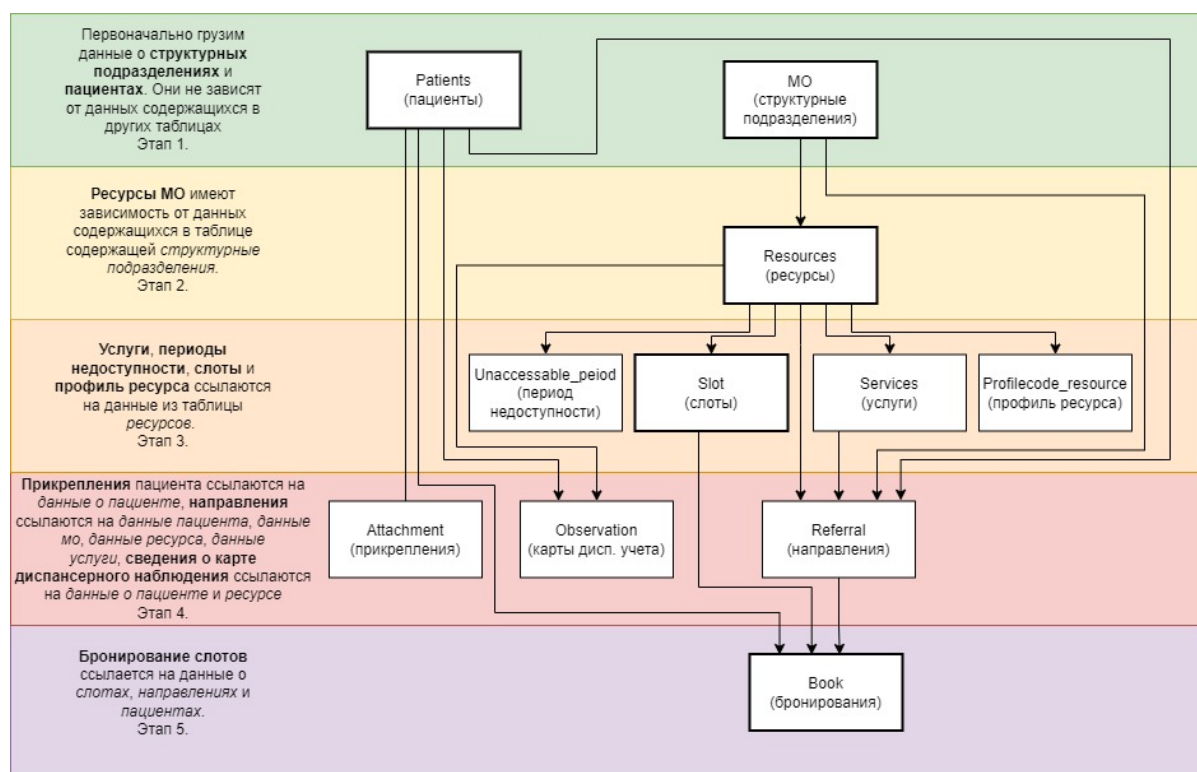


Рисунок 4 - Порядок выгрузки данных

3.1.4 Требования к наполнению данными

Наполнение «Региональной витрины данных субъекта РФ» данными, необходимыми для надлежащего оказания услуги запись на прием к врачу, должно выполняться в соответствии со следующими требованиями:

1. Периодичность наполнения / обновления:

данные о расписании МО (сведения о свободных и занятых слотах, таблица slot) – не реже чем раз в 30 сек;

прочие данные (сведения о ресурсах МО, пациентах, направлениях и прочее) – не реже чем раз в 1 час.

При изменении статуса слота через другие источники также необходимо в течение 30 сек отправить данные с актуальным слотом.

3.1.5 Требование к удалению данных

На Витрине данных реализовано каскадное удаление, при удалении родительского элемента также удаляются все связанные с ним дочерние элементы:

- при удалении МО, удаляются ресурсы, слоты, бирки, услуги, прикрепления;
- при удалении ресурса, удаляются слоты, бирки, услуги;
- при удалении слота, удаются бирки.

3.2 Описание сценария «Запись на прием к врачу»

В данном разделе описывается сценарий «Запись на прием к врачу».

3.2.1 Участники

Пользователи:

- Пользователь.

ИС:

- ЕПГУ.
- Компонент «Витрина данных «Запись к врачу».
- ФЭР шлюз.
- МО.
- ЕМИС (Единая медицинская информационная система)
- Kafka (промежуточная очередь сообщений)

3.2.2 Критерий успешности выполнения

Пользователь записан на прием к врачу в МО.

Предварительные условия

- 1) Пользователь авторизован посредством ЕСИА.
- 2) Пользователем выбрана услуга «Запись на прием к врачу».

3.2.3 Базовый сценарий

Сценарий предоставления пользователю услуги состоит из двух частей:

- 1) Предоставление информации из ЕМИС пользователю – реализуется с использованием регламентированных запросов.
- 2) Осуществление изменения информации в ЕМИС по инициативе пользователя – реализуется с использованием спецификаций OpenAPI.

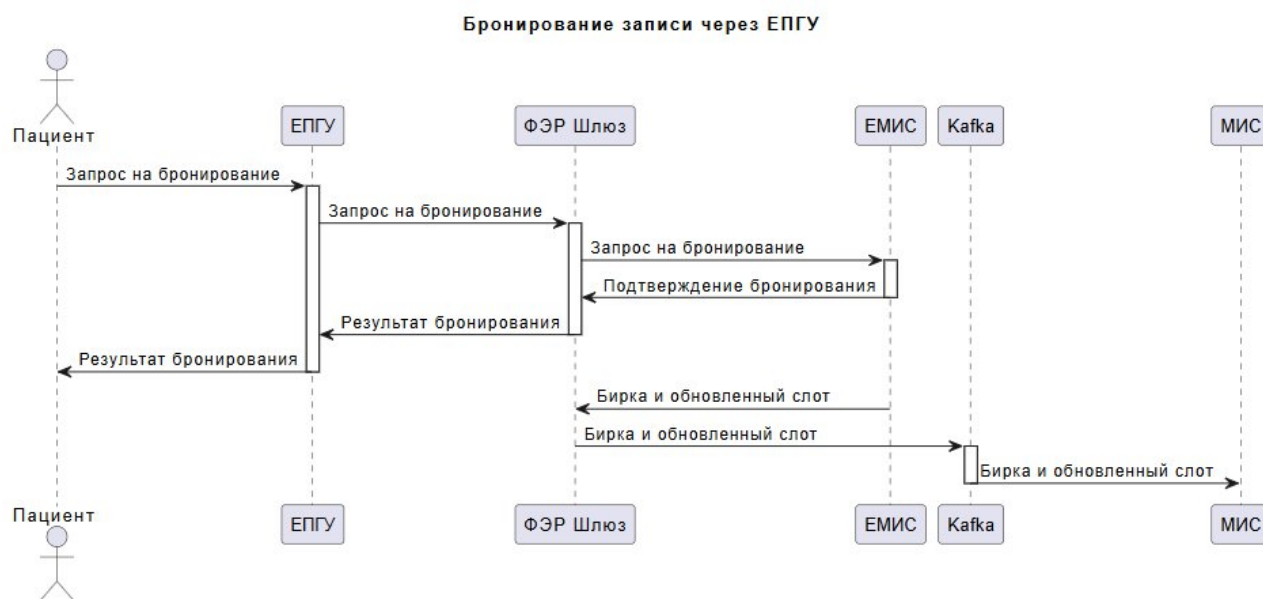


Рисунок 5 – Диаграмма процесса предоставления пользователю услуги «Запись на прием к врачу»

Информационный обмен осуществляется следующим образом (диаграмма последовательности приведена на рисунке 5):

1. Пациент заходит на ЕПГУ и подает запрос на бронирование записи.
2. ЕПГУ передает запрос на бронирование в "ФЭР Шлюз".
3. "ФЭР Шлюз" пересылает запрос в ЕМИС.
4. ЕМИС обрабатывает запрос и подтверждает бронирование.
5. "ФЭР Шлюз" передает результат бронирования обратно в ЕПГУ.
6. ЕПГУ сообщает Пациенту о результате бронирования.
7. Параллельно с п. 3, ЕМИС отправляет в "ФЭР Шлюз" данные о бирке и обновленном слоте.
8. "ФЭР Шлюз" пересылает эти данные в Kafka.
9. Kafka доставляет данные в МИС.
10. МИС обновляет свои данные на основе полученной информации.

3.3 Описание сценария «Отмена записи на прием к врачу»

В данном разделе описывается сценарий «Отмена записи на прием к врачу».

3.3.1 Участники

Пользователи:

- Пользователь.

- ЕПГУ.
- Компонент «Витрина данных «Запись к врачу».
- ФЭР шлюз.
- МО.
- ЕМИС (Единая медицинская информационная система)
- Kafka (промежуточная очередь сообщений)

3.3.2 Критерий успешности выполнения

Пользователь отменил запись на прием к врачу в МО.

Предварительные условия

- 1) Пользователь авторизован посредством ЕСИА.
- 2) Пользователь перешел в раздел Заявления в ЛК и выбрал запись, которую хочет отменить.

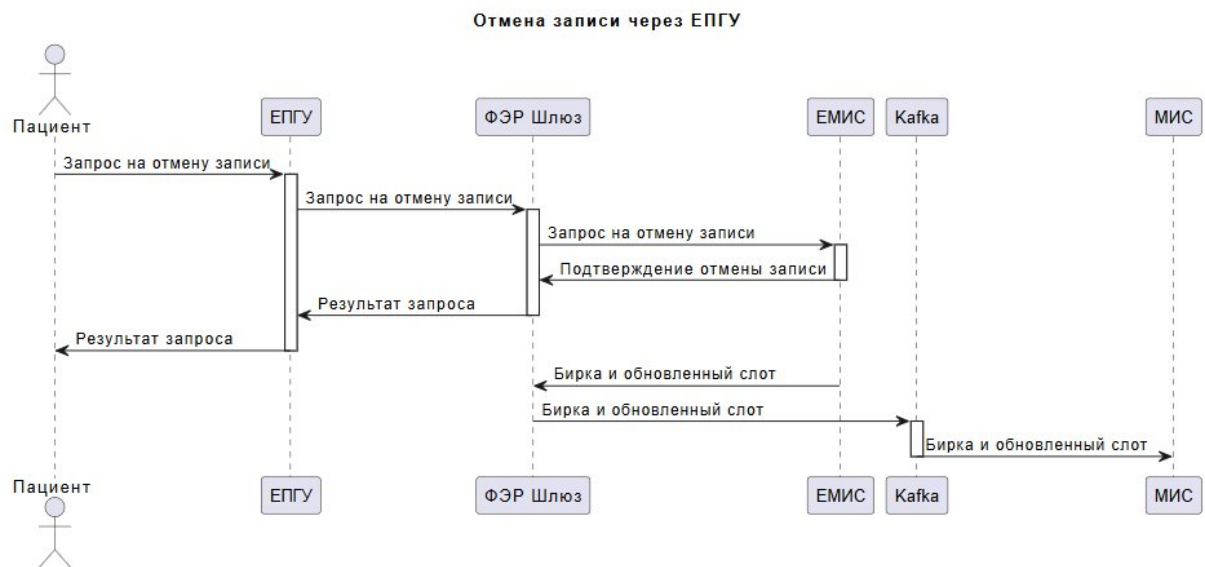


Рисунок 6 – Диаграмма процесса предоставления пользователю услуги «Отмена записи»

3.3.3 Сценарий отмены (описание последовательности шагов)

1. Пациент обращается к ЕПГУ с запросом на отмену записи.
2. ЕПГУ передает запрос на отмену в "ФЭР Шлюз".
3. "ФЭР Шлюз" направляет запрос ЕМИС.
4. ЕМИС обрабатывает запрос и подтверждает отмену записи.
5. "ФЭР Шлюз" передает результат запроса обратно в ЕПГУ.
6. ЕПГУ информирует Пациента о результате запроса.

7. Параллельно с (п. 3), после получения запроса на отмену записи ЕМИС отправляет в "ФЭР Шлюз" обновленные данные о бирке и слоте.
8. "ФЭР Шлюз" перенаправляет эти данные в Kafka.
9. Kafka доставляет данные МИС.
10. МИС обновляет свои данные на основе полученной информации.

4 Сервис авторизации

Каждый запрос к API проходит авторизацию, в ходе которой сверяется значение параметра value со значением на сервере авторизации. Для получения value необходимо выполнить запрос к сервису авторизации.

Адрес	https://<адрес сервера>/auth.svc
Метод	POST
Входные данные	См. Error: Reference source not found
Пример запроса	<pre>POST /auth.svc HTTP/1.1 Host: https://covid-test.miacugra.ru Content-Type: application/json User-Agent: PostmanRuntime/7.15.2 Accept: */* Cache-Control: no-cache Postman-Token: af098560-3f4d-47e8-8694-d1e514e5843f,717aabe7-09f9-4a03-b10b-4e1247d963dd Host: 192.168.2.126:1105 Accept-Encoding: gzip, deflate Content-Length: 72 Connection: keep-alive cache-control: no-cache { "username": " 1.2.643.5.1.13.13.12.2.86.8902", "password": "E837EE78350021618250663B60EF58**" }</pre>
Пример ответа	<pre>{ "Result": { "ValidTo": "2019-10-08T19:12:19Z", "Value": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJNTyBJbnRlZ3JhdGlvbiIsImp0aSI6ImQ2MTc5ZmRiLTJlTFJlN2EtNGZjYS05ZDc1LTAwNzRjZjA0Y2IzOCIsIklvSWQ9IiwiaXNjaW50IjoiImV4cCI6MTU3MDU2MTkzOSwiaXNzIjoivGVsZW1lZEFwaSIsImF1ZCI6ImlRbGVtZW50ZW50ZW50ZW50LjQ0OD86Zl40UvJWEGF1SjWFirN9bvIgT-91qQXEafByKE" }, "IsSuccess": true, "ErrorMessage": "", "StatusCode": 200 }</pre>

Таблица 1 – Формат объекта «Авторизация»

Наименование	Тип	Описание
username	String	Oid медицинского учреждения. Справочник OID 1.2.643.5.1.13.13.11.1461
password	String	Пароль

Для получения value необходимо отправить HTTP POST запрос на адрес сервиса, в теле запроса необходимо отправить запрос с типом application/json если указывать тип с двумя параметрами: username, password.

Полученное значение value необходимо использовать в дальнейшем при обращении к API. Для каждого запроса необходимо добавлять заголовок:

Authorization: bearer <access_token>, где access_token - значение токена полученное при успешной авторизации. При истечении действия токена необходимо обновить токен авторизации.

4.1 Авторизация к топикам Kafka

Система	SecurityProtocol	SaslMechanism	GroupId	Список топиков и прав
ЕМИС	SaslPlaintext	Plain	emis_consumerGroup	emis_attachment (READ) emis_book (READ) emis_home_visit (READ) emis_locations (READ) emis_medcase (READ) emis_medvisit (READ) emis_mo (READ) emis_observation (READ) emis_patient (READ) emis_profilecode_resource (READ) emis_referral (READ) emis_resource_location (READ) emis_resource (READ) emis_service (READ) emis_slot (READ) emis_unaccessible_period (READ) emis_waiting_list (READ) emis_waiting_list_entity (READ)

5 Сервис взаимодействия с программным обеспечением «Витрина данных»

Методы сервиса взаимодействия с ПО «Витрина данных» используются для предоставления сведений о пользователе, медицинских организациях, медицинских работниках, свободных слотах для записи на прием к врачу и информации о текущих записях.

Общие сведения о методах и описание входных и выходных данных методов приведено в таблицах ниже.

Дополнение:

Запросы, отправляемые к подсистемам, должны быть сжаты с использованием gzip, для экономии сетевого трафика и ускорения взаимодействия. В заголовке запроса, передаваемого с использованием gzip, необходимо обязательно указывать атрибуты:

Таблица 2 – Обязательные атрибуты заголовка запроса при использовании gzip

Описание	Наименование	Значение
Способ кодирования содержимого сообщения	Content-Encoding	gzip
Определение типа содержимого	Content-Type	application/json

На переходный период поддержка приёма запросов без сжатия gzip не прекращается.

5.1 Описание подсистемы МО

Методы данной подсистемы необходимо использовать при передаче, обновлении и удалении данных структурных подразделений медицинских организаций.

Добавлены проверки при отправке подразделений:

1. Проверка на дублирование наименования подразделения. При отправке МО, будет осуществляться проверка на совпадение наименования с уже имеющимися записями в Витрине данных.

Запрос:

```
{
  "data": [
    {
      "id": "1b0aaec1-38ba-4bf2-a82c-39f90b925a78",
      "parent_id": "1.2.643.5.1.13.13.12.2.86.8916",
      "name": "Тестовая МО",
      "address": "ХМАО-Югра, г. Нефтеюганск, 10 микрорайон, дом 1",
      "address_fias_guid": "string",
      "region_okato": "string",
      "phone": "string",
      "enabled": "NONE",
      "create_ts": "20.10.2023 08:00:00",
      "update_ts": "string",
      "branch_oid": "1.2.643.5.1.13.13.12.2.86.8964.0.163227"
    }
  ]
}
```

Ответ:

```
{
  "isSuccess": false,
  "error": {
    "errorMessage": "Структурное подразделение с идентификатором [1b0aaec1-38ba-4bf2-a82c-39f90b925a78] дублирует ранее загруженное в витрину структурное подразделение с таким наименованием [name] и с другим идентификатором [6b0aaec1-38ba-4bf2-a82c-39f90b925a78]. [1b0aaec1-38ba-4bf2-a82c-39f90b925a78].",
    "errorCode": "R-310"
  },
  "errors": [
    {
      "id": "1b0aaec1-38ba-4bf2-a82c-39f90b925a78",
      "errorMessage": "Структурное подразделение с идентификатором [1b0aaec1-38ba-4bf2-a82c-39f90b925a78] дублирует ранее загруженное в витрину структурное подразделение с таким наименованием [name] и с другим идентификатором [6b0aaec1-38ba-4bf2-a82c-39f90b925a78].",
      "errorCode": "R-330"
    }
  ],
  "sendedGuids": [],
  "sendedGuidsToContactCenter": null,
  "statusCode": 422
}
```

5.1.1 Добавление и обновление данных

Адрес	http://<адрес сервера>/api/mo
Метод	POST
Входные данные	Полное описание объекта приведено ниже (Приложение 1. Таблица 1 - МО)
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.1.2 Получение данных подразделения по идентификатору

Адрес	http://<адрес сервера>/api/mo/{id}
Метод	GET
Входные данные	id – уникальный идентификатор записи
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных») Описание результата ответа приведено ниже (Приложение 1. Таблица 1 - МО)

5.1.3 Получение данных о подразделениях МО по OID МО

Адрес	http://<адрес сервера>/api/oid/{oid}
Метод	GET
Входные данные	oid – OID МО Пример запроса: GET /api/mo/oid/1.2.643.5.1.13.13.12.2.86.8999
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных») Описание результата ответа приведено ниже (Приложение 1. Таблица 1 - МО)

5.1.4 Удаление данных

Адрес	http://<адрес сервера>/api/mo
Метод	DELETE
Входные данные	id – уникальный идентификатор записи
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.1.5 Получение данных о подразделениях МО доступных для записи по ID пациента

Адрес	http://<адрес сервера>/api/data
Метод	POST
Входные данные	patientId – уникальный идентификатор пациента mo_oid – OID головной медицинской организации, к которой прикреплен пациент Пример запроса: <pre>POST /api/mo/data?patientId={patientId} Host: Content-Type: application/json Authorization: Content-Length: 40 ["{oid}"]</pre>
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.2 Описание подсистемы Resource

Методы данной подсистемы необходимо использовать при передаче, обновлении и удалении данных ресурсов медицинской организации.

5.2.1 Добавление и обновление данных

Адрес	http://<адрес сервера>/api/resource
Метод	POST
Входные данные	Полное описание объекта приведено ниже (Приложение 1. Таблица 2 - Resource)
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.2.2 Получение данных

Адрес	http://<адрес сервера>/api/resource/{id}
Метод	GET
Входные данные	id – уникальный идентификатор записи
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных») Описание результата ответа приведено ниже (Приложение 1. Таблица 2 - Resource)

5.2.3 Получение данных ресурсов по МО

Адрес	http://<адрес сервера>/api/mo/{id}/resources
Метод	GET
Входные данные	id – уникальный идентификатор МО
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных») Описание результата ответа приведено ниже (Приложение 1. Таблица 2 - Resource)

5.2.4 Удаление данных

Адрес	http://<адрес сервера>/api/resource
Метод	DELETE
Входные данные	id – уникальный идентификатор записи
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.2.5 Получение доступных должностей по МО

Адрес	http://<адрес сервера>/api/mo/{id}/available-posts
Метод	GET
Входные данные	id – уникальный идентификатор МО patient-id – уникальный идентификатор пациента (опционально, query) tag-types – тип записей слотов (EPGU, ORDINARY и т.д) (опционально, query)

	with_busy_slots – true / false, в зависимости от переданного значения возвращает только свободные или все слоты, включая занятые epgu-list – true/false (в зависимости от переданного параметра будут возвращаться только специальности доступные через ЕПГУ или полный список специальностей) tag-age – выводит специальности в зависимости от возраста пациента (ADULT – взрослый (с 18 лет); CHILD – детский (с 1 года до 17 лет вкл.); BABY – младенец (с 0 до 12 мес. вкл.); ALL – все возрастные группы) если передан ADULT – скрываются детские специальности, иное – выводятся все специальности) Пример запроса <pre>GET /mo/{mo-id}/available-posts?patient-id={patientId}&tag-types={tag-type}&with_busy_slots={true/false}&epgu-list={bool}&tag-age={age} Host: Accept: text/plain Authorization:</pre>
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.3 Описание подсистемы Service

Методы данной подсистемы необходимо использовать при передаче, обновлении и удалении услуги

5.3.1 Добавление и обновление данных

Адрес	http://<адрес сервера>/api/service
Метод	POST
Входные данные	Полное описание объекта приведено ниже (Приложение 1. Таблица 3 – Service)
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.3.2 Получение данных

Адрес	http://<адрес сервера>/api/service
Метод	GET

Входные данные	id – уникальный идентификатор записи
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»). Описание результата ответа приведено ниже (Приложение 1. Таблица 3 – Service).

5.3.3 Удаление данных

Адрес	http://<адрес сервера>/api/service
Метод	DELETE
Входные данные	id – уникальный идентификатор записи
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.4 Описание подсистемы Slot

Методы данной подсистемы необходимо использовать при передаче, обновлении и удалении слотов.

Добавлены проверки при отправке слотов:

1. Проверка на дублирование идентификаторов слота (id) **в рамках одного запроса.**

Пример запроса с ошибкой:

```
{
  «data»: [
    {
      «id»: «aaaaaaa1-5eff-49b8-9d07-ff2486162b3f»,
      «mo_id»: «ab7cbc4b-2e68-40d2-881b-6f957b044818»,
      «resource_id»: «1919135c-7c9e-4564-a8b4-29aff7adfe72»,
      «tag_type»: «EPGU»,
      «tag_age»: «ALL»,
      «visit_time»: «2024-03-28T05:00:00»,
      «duration»: «20»,
      «create_ts»: «2023-10-01 09:00:00»,
      «status»: «AVAILABLE»
    },
    {
      «id»: «aaaaaaa1-5eff-49b8-9d07-ff2486162b3f»,
      «mo_id»: «ab7cbc4b-2e68-40d2-881b-6f957b044818»,
      «resource_id»: «436b6461-efb-4f3c-ad6f-27f4b2c3e578»,
      «tag_type»: «EPGU»,
      «tag_age»: «ALL»,
      «visit_time»: «2024-03-29T05:00:00»,
      «duration»: «20»,
      «create_ts»: «2023-10-01 09:00:00»,
      «status»: «AVAILABLE»
    }
  ]
}
```

Ответ:

```
{
  «isSuccess»: false,
  «error»: {
    «errorMessage»: «В слоте с идентификатором 'aaaaaaa1-5eff-49b8-9d07-ff2486162b3f':  
Уникальный идентификатор слота [id] имеет дубликат внутри запроса»,
    «errorCode»: «R-310»
  },
  «errors»: [
    {
      «id»: «aaaaaaa1-5eff-49b8-9d07-ff2486162b3f»,
      «errorMessage»: «В слоте с идентификатором 'aaaaaaa1-5eff-49b8-9d07-ff2486162b3f': Уникальный  
идентификатор слота [id] имеет дубликат внутри запроса»,
      «errorCode»: «R-310»
    }
  ],
  «sendedGuids»: [
```

```

"aaaaaaa1-5eff-49b8-9d07-ff2486162b3f"
],
"sendedGuidsToContactCenter": [
  "aaaaaaa1-5eff-49b8-9d07-ff2486162b3f"
],
«statusCode»: 422}

```

2. Проверка на дублирование слотов по времени начала приема (visit_time) и идентификатору ресурса (resource_id). При отправке слота, будет осуществляться проверка на совпадение времени начала приема и врача с уже имеющимися записями в Витрине данных.

Пример запроса:

```

{
  «data»: [
    {
      «id»: «aaaaaaa2-5eff-49b8-9d07-ff2486162b3f»,
      «mo_id»: «ab7cbc4b-2e68-40d2-881b-6f957b044818»,
      «resource_id»: «1919135c-7c9e-4564-a8b4-29aff7adfe72»,
      «tag_type»: «EPGU»,
      «tag_age»: «ALL»,
      «visit_time»: «2024-03-28T05:00:00»,
      «duration»: «20»,
      «create_ts»: «2023-10-01 09:00:00»,
      «status»: «AVAILABLE»
    }
  ]
}

```

Ответ:

```

{
  «isSuccess»: false,
  «error»: {
    «errorMessage»: «Слот с идентификатором [aaaaaaa2-5eff-49b8-9d07-ff2486162b3f] и указанным в нем ресурсом [resource_id] и временем визита [visit_time] дублирует ранее загруженный в витрину слот с другим идентификатором [aaaaaaa1-5eff-49b8-9d07-ff2486162b3f]. [aaaaaaa2-5eff-49b8-9d07-ff2486162b3f].»,
    «errorCode»: «R-310»
  },
  «errors»: [
    {
      «id»: «aaaaaaa2-5eff-49b8-9d07-ff2486162b3f»,
      «errorMessage»: «Слот с идентификатором [aaaaaaa2-5eff-49b8-9d07-ff2486162b3f] и указанным в нем ресурсом [resource_id] и временем визита [visit_time] дублирует ранее загруженный в витрину слот с другим идентификатором [aaaaaaa1-5eff-49b8-9d07-ff2486162b3f].»,
      «errorCode»: «R-330»
    }
  ],
  «sendedGuids»: [],
  «sendedGuidsToContactCenter»: [],
  «statusCode»: 422
}

```

5.4.1 Добавление и обновление данных

Адрес	http://<адрес сервера>/api/slot
Метод	POST

Входные данные	Полное описание объекта приведено ниже (Приложение 1. Таблица 4 – Slot)
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.4.2 Получение данных

Адрес	http://<адрес сервера>/api/slot/{id}
Метод	GET
Входные данные	id – уникальный идентификатор записи
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных») Описание результата ответа приведено ниже (Приложение 1. Таблица 4 – Slot)

5.4.3 Получение данных слотов по ресурсу

Адрес	http://<адрес сервера>/api/resource/{id}/slots
Метод	GET
Входные данные	id – уникальный идентификатор ресурса (врача) startDate – начало интервала поиска по времени слота. Формат YYYY-MM-DD (опционально, query) endDate – конец интервала поиска по времени слота. Формат YYYY-MM-DD (опционально, query) Пример запроса: <pre>GET /api/resource/{id}/slots?startDate=YYYY-MM-DD&endDate=YYYY-MM-DD Host: Accept: text/plain Authorization:</pre>
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных») Описание результата ответа приведено ниже (Приложение 1. Таблица 4 – Slot)

5.4.4 Получение слотов по OID МО и коду должности

Адрес	http://<адрес сервера>/api/mo/resources/slots
Метод	GET

Входные данные	mo-oid (обязательно) – OID МО, для которого требуется получить информацию о слотах (query) resources-post-id (обязательно) – идентификатор кода должности ресурса (query) slot-status – статус слота (например, «AVAILABLE») (опционально, query) tag_types – тип слота (например, «EPGU») (опционально, query) Пример запроса: <pre>GET /api/mo/resources/slots?mo-oid={oid}&resources-post-id={post-id}&slot-status={status}&tag-types={type} Host: Accept: text/plain Authorization:</pre>
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»). Описание результата ответа приведено ниже (Приложение 1. Таблица 4 – Slot)

5.4.5 Получение слотов по МО и коду должности ресурса

Адрес	http://<адрес сервера>/api/mo/{id}/resources/slots
Метод	GET
Входные данные	id (обязательно) – id МО, для которого требуется получить информацию о слотах (query) resources-post-id (обязательно) – идентификатор кода должности ресурса (query) slot-status – статус слота (например, «AVAILABLE») (опционально, query) tag-types – тип слота (например, «EPGU») (опционально, query) includeExpired – true/false – возвращает включая/исключая устаревшие слоты tag-age – возраст пациента, выводятся слоты с соответствующим значением в поле tag_age (ADULT – взрослый (с 18 лет); CHILD – детский (с 1 года до 17 лет вкл.); BABY – младенец (с 0 до 12 мес. вкл.); ALL – все возрастные группы) Пример запроса: <pre>GET /api/mo/{id}/resources/slots?mo-oid={oid}&resources-post-id={post-id}&slot-status={status}&tag-types={type}&includeExpired={true/false}&tag-age={age} Host: Accept: text/plain Authorization:</pre>
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»). Описание результата ответа приведено ниже (Приложение 1. Таблица 4 – Slot)

5.4.6 Удаление данных

Адрес	http://<адрес сервера>/api/slot
Метод	DELETE
Входные данные	id – уникальный идентификатор записи
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.5 Описание подсистемы Patient

Данные по пациенту выгружаются через Региональный регистр пациентов.

5.6 Описание подсистемы Attachment

Данные по прикреплениям выгружаются через Региональный регистр пациентов.

5.7 Описание подсистемы Book

Методы данной подсистемы необходимо использовать при передаче, обновлении и удалении бирки.

5.7.1 Добавление и обновление данных

Адрес	http://<адрес сервера>/api/book
Метод	POST
Входные данные	Полное описание объекта приведено ниже (Приложение 1, Таблица 7 – Book)
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

Перед выгрузкой бирки, необходимо выгрузить слот со статусом RECORDED. В случае если для выгружаемой бирки не будет найден слот, возвращается ошибка и бирка не выгружается в Витрину.

Если бронирование записи производилось через ЕПГУ, то при выгрузке бирки проверяется соответствие id бирки и book_ext_id.

В случае если, id бирки и book_ext_id не совпадают возвращается ошибка, бирка не выгружается.

5.7.2 Получение данных

Адрес	http://<адрес сервера>/api/book/{id}
Метод	GET
Входные данные	id – уникальный идентификатор записи
Выходные данные	Полное описание объекта приведено ниже (Приложение 1. Таблица 7 - Book)

5.7.3 Получение данных бирок по идентификатору слота

Адрес	http://<адрес сервера>/api/slot/{id}/books
Метод	GET
Входные данные	slot_id – уникальный идентификатор слота
Выходные данные	Полное описание объекта приведено ниже (Приложение 1. Таблица 7 - Book)

5.7.4. Удаление данных

Адрес	http://<адрес сервера>/api/book
Метод	DELETE
Входные данные	id – уникальный идентификатор записи
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.7.5 Получение бирок пациента

Адрес	http://<адрес сервера>/api/books
Метод	GET
Входные данные	patient_id – уникальный идентификатор пациента status_codes – статус записи (integer, 5000 активная запись, 5010 – отмененная запись и т.д.) types – тип записи (напр. APPOINTMENT и т.д.) Пример запроса

	GET /api/books?patient_id={id}&status_codes={code}&types={type} Host: Accept: Authorization:
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.8 Описание подсистемы Referral

Методы данной подсистемы необходимо использовать при передаче, обновлении и удалении направления пациентов.

5.8.1 Добавление и обновление данных

Адрес	http://<адрес сервера>/api/referral
Метод	POST
Входные данные	Полное описание объекта приведено ниже (Приложение 1, Таблица 8 – Referral)
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.8.2 Получение данных

Адрес	http://<адрес сервера>/api/referral/{id}
Метод	GET
Входные данные	id – уникальный идентификатор записи
Выходные данные	Полное описание объекта приведено ниже (Приложение 1. Таблица 8 – Referral)

5.8.3 Удаление данных

Адрес	http://<адрес сервера>/api/referral
Метод	DELETE
Входные данные	id – уникальный идентификатор записи

Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)
------------------------	---

5.8.4 Получение данных направления по номеру направления

Адрес	http://<адрес сервера>/api/referral/number/{number}
Метод	GET
Входные данные	number – номер направления
Выходные данные	Полное описание объекта приведено ниже (Приложение 1. Таблица 8 – Referral)

5.8.5 Получение данных направления по идентификатору пациента

Адрес	http://<адрес сервера>/api/referral/patient/{id}
Метод	GET
Входные данные	id – уникальный идентификатор пациента
Выходные данные	Полное описание объекта приведено ниже (Приложение 1. Таблица 8 – Referral)

5.9 Описание подсистемы Unaccessible_period

Методы данной подсистемы необходимо использовать при передаче, обновлении и удалении периодов недоступности ресурсов.

5.9.1 Добавление и обновление данных

Адрес	http://<адрес сервера>/api/unaccessible_period
Метод	POST
Входные данные	Полное описание объекта приведено ниже (Приложение 1, Таблица 9 – Unaccessible_period)
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

Период недоступности должен быть привязан к сущности «Врач» и выгружаться одной записью, охватывающей весь планируемый период отсутствия врача

5.9.2 Получение данных

Адрес	http://<адрес сервера>/api/unaccessible_period/{id}
Метод	GET
Входные данные	id – уникальный идентификатор записи
Выходные данные	Полное описание объекта приведено ниже (Приложение 1. Таблица 9 – Unaccessible_period)

5.9.3 Удаление данных

Адрес	http://<адрес сервера>/api/unaccessible_period
Метод	DELETE
Входные данные	id – уникальный идентификатор записи
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.9.4. Получение данных о периодах недоступности по идентификатору ресурса

Адрес	http://<адрес сервера>/api/unaccessible_period/resource/{id}
Метод	GET
Входные данные	id – уникальный идентификатор ресурса (врача)
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.10 Описание подсистемы Profilecode_resource

Методы данной подсистемы необходимо использовать при передаче, обновлении и удалении данных профиля.

5.10.1 Добавление и обновление данных

Адрес	http://<адрес сервера>/api/profilecode_resource
Метод	POST
Входные данные	Полное описание объекта приведено ниже (Приложение 1, Таблица 10 – Profilecode_resource)
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.10.2 Получение данных

Адрес	http://<адрес сервера>/api/profilecode_resource/{id}
Метод	GET
Входные данные	id – уникальный идентификатор записи
Выходные данные	Полное описание объекта приведено ниже (Приложение 1. Таблица 10 – Profilecode_resource)

5.10.3 Удаление данных

Адрес	http://<адрес сервера>/api/profilecode_resource
Метод	DELETE
Входные данные	id – уникальный идентификатор записи
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.11 Описание подсистемы Waiting_list

Методы данной подсистемы необходимо использовать при передаче, обновлении и удалении листов ожидания

5.11.1 Добавление и обновление данных

Адрес	http://<адрес сервера>/api/waiting_list
Метод	POST
Входные данные	Полное описание объекта приведено ниже (Приложение 1, Таблица 11 – Waiting_list)
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.11.2 Получение данных

Адрес	http://<адрес сервера>/api/waiting_list/{id}
Метод	GET
Входные данные	id – уникальный идентификатор записи
Выходные данные	Полное описание объекта приведено ниже (Приложение 1, Таблица 11 – Waiting_list)

5.11.3 Удаление данных

Адрес	http://<адрес сервера>/api/waiting_list
Метод	DELETE
Входные данные	id – уникальный идентификатор записи
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.12 Описание подсистемы Waiting list entity

5.12.1 Добавление и обновление данных

Адрес	http://<адрес сервера>/api/waiting_list_entity
Метод	POST
Входные данные	Полное описание объекта приведено ниже (Приложение 1, Таблица 12 – Waiting_list_entity)
Выходные дан-	Пример описания ответа приведено ниже (Приложение 2. Структура

ные	ответа сервиса взаимодействия с ПО «Витрина данных»)
------------	--

5.12.2 Удаление данных

Адрес	http://<адрес сервера>/api/waiting_list_entity
Метод	DELETE
Входные данные	id – уникальный идентификатор записи
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.13 Описание подсистемы Home Visit

Методы данной подсистемы необходимо использовать при передаче, обновлении и удалении вызовов врачей на дом.

5.13.1 Добавление и обновление данных

Адрес	http://<адрес сервера>/api/home_visit
Метод	POST
Входные данные	Полное описание объекта приведено ниже (Приложение 1, Таблица 13 – Home visit)
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.13.2 Удаление данных

Адрес	http://<адрес сервера>/api/home_visit
Метод	DELETE
Входные данные	id – уникальный идентификатор записи
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.13.3 Получение данных о вызове врача на дом

Адрес	http://<адрес сервера>/api/home_visit/{id}
Метод	GET
Входные данные	id – уникальный идентификатор записи (вызова)
Выходные данные	Полное описание объекта приведено ниже (Приложение 1. Таблица 13 – Home_Visit)

5.13.4 Добавление адресов вызовов врачей на дом

Адрес	http://<адрес сервера>/api/home_visit/addresses
Метод	POST
Входные данные	Полное описание объекта приведено ниже (Приложение 1, Таблица 13 – Home visit)
Выходные данные	Полное описание объекта приведено ниже (Приложение 1. Таблица 13 – Home_Visit)

5.14 Описание подсистемы Locations

5.14.1 Добавление и обновление данных

Адрес	http://<адрес сервера>/api/locations
Метод	POST
Входные данные	Полное описание объекта приведено ниже (Приложение 1, Таблица 14 – Locations)
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.14.2 Получение данных об участке медицинской организации

Адрес	http://<адрес сервера>/api/locations/{id}
Метод	GET

Входные данные	id – уникальный идентификатор участка
Выходные данные	Полное описание объекта приведено ниже (Приложение 1. Таблица 14 – Locations)

5.14.3 Получение данных об участках медицинской организации по идентификатору МО

Адрес	http://<адрес сервера>/api/mo/{id}/locations
Метод	GET
Входные данные	id – уникальный идентификатор МО
Выходные данные	Полное описание объекта приведено ниже (Приложение 1. Таблица 14 – Locations)

5.14.4 Удаление данных

Адрес	http://<адрес сервера>/api/locations
Метод	DELETE
Входные данные	id – уникальный идентификатор записи
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.15 Описание подсистемы Resource Location

5.15.1 Добавление и обновление данных

Адрес	http://<адрес сервера>/api/resource_location
Метод	POST
Входные данные	Полное описание объекта приведено ниже (Приложение 1, Таблица 15 – Resource_Locations)
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.15.2 Получение данных о ресурсе закрепленным за участком

Адрес	http://<адрес сервера>/api/resource_location/{id}
Метод	GET
Входные данные	id – уникальный идентификатор прикрепления ресурса (врача)
Выходные данные	Полное описание объекта приведено ниже (Приложение 1. Таблица 15 – Resource_Locations)

5.15.3 Получение данных о ресурсах, закрепленных за участками по идентификатору ресурса

Адрес	http://<адрес сервера>/api/resource/{id}/resource_locations
Метод	GET
Входные данные	id – уникальный идентификатор ресурса (врача)
Выходные данные	Полное описание объекта приведено ниже (Приложение 1. Таблица 15 – Resource_Locations)

5.15.4 Удаление данных

Адрес	http://<адрес сервера>/api/resource_locations
Метод	DELETE
Входные данные	id – уникальный идентификатор записи
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.16 Описание подсистемы Observation

5.16.1 Добавление и обновление данных

Адрес	http://<адрес сервера>/api/observation
--------------	--

Метод	POST
Входные данные	Полное описание объекта приведено ниже (Приложение 1, Таблица 16 – Observation)
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.16.2 Удаление данных

Адрес	http://<адрес сервера>/api/observation
Метод	DELETE
Входные данные	id – уникальный идентификатор записи
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

5.16.3 Получение данных

Адрес	http://<адрес сервера>/api/observation/{id}
Метод	GET
Входные данные	id – уникальный идентификатор записи
Выходные данные	Пример описания ответа приведено ниже (Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»)

6 Сервис бронирования и отмены записи на прием

Для осуществления записи на прием в адаптере бронирования существуют следующие эндпоинты:

ContactCenter/booking – используется ЕМИС, когда источником подтверждения бронирования выступает МИС

MIS/booking – используется МИС, когда источником подтверждения бронирования выступает ЕМИС

Оба эндпоинта имеют в своем составе методы на создание бронирования и на его отмену, описанные в пунктах 6.1 и 6.2.

Диаграммы последовательностей изображены на рисунках 7, 8.

6.1 Метод бронирования записи

Метод POST /booking/book используется для бронирования слота. ФЭР шлюз передает набор данных в ЕМИС / МИС для бронирования, в ответном сообщении ЕМИС / МИС возвращает извещение о результатах обработки запроса.

Метод должен обрабатывать не более 10 секунд.

Общие сведения о методе и описание входных и выходных данных метода приведено в таблицах ниже.

Описание параметров в запросе:

№	Параметр	Тип данных	Обязательность	Описание
1	slot_id	string	О	Идентификатор слота GUID
2	referral_id	string	У	Идентификатор направления, при записи по направлению. Обязательно при записи по направлению
3	preliminary_reservation	boolean	Н	Переключатель на бронирование/предварительное бронирование true/false (по умолчанию всегда false)
4	patient_id	string	О	Идентификатор пациента GUID
5	mobile_phone	string	Н	Номер телефона пациента из ЕПГУ
6	mo_oid	string	Н	Идентификатор медицинской организации (МО) OID
7	email	string	Н	Электронная почта пациента из ЕПГУ
8	depart_oid	string	Н	Идентификатор подразделения
9	cards_id	string	Н	Идентификатор медицинской карты
10	booking_type	string	О	Тип записи (APPOINTMENT /VACCINATION /D_OBSERVATION /ONLINE_CONSULTING)
11	booking_source	string	Н	Источник записи. Возможные значения: INFOMAT, DOCTOR, OTHER_WEB

Пример запроса:

```
{
  "booking_type": "APPOINTMENT",
  "slot_id": "3aa0bd15-39e4-42f9-8a2e-a85a04d011cf",
  "patient_id": "80ea05f3-5cfd-4b7b-8acb-f2c31323d78f",
  "mo_oid": "1.2.643.5.1.13.13.12.2.86.8892",
```

```

"depart_oid": "1.2.643.5.1.13.13.12.2.86.8892.0.150193",
"preliminary_reservation": false,
"referral_id": null,
"cards_id": null,
"email": "patient@example.com",
"mobile_phone": "+79998887766",
"booking_source": "DOCTOR"
}

```

Описание параметров в ответе:

№	Параметр	Тип данных	Обязательность	Описание
1	book_ext_id	string	О	Идентификатор записи на стороне РМИС Данный идентификатор необходимо использовать в качестве id записи при выгрузке бирки (Таблица 7 – Book - записи на бирку пациентов, поле – id)
2	patient_id	string	О	Идентификатор пациента GUID
3	slot_id	string	О	Идентификатор слота
4	mo_oid	string	Н	Идентификатор медицинской организации (МО) OID
5	depart_oid	string	Н	Идентификатор подразделения
6	booking_source	string	Н	Источник записи. Возможные значения: INFOMAT, DOCTOR, OTHER_WEB

Пример ответа:

```

{
  "book_ext_id": "b3312d6e-5eff-49b8-9d07-ff2486162b2f",
  "slot_id": "b3312d6e-5eff-49b8-9d07-ff2486162b2f",
  "type": "BookResponseSuccess",
  "status": {
    "statusCode": 200,
    "statusMessage": "Бронирование успешно завершено"
  }
}

```

6.1.1 Оптимизация метода бронирования записи

Сценарий:

Название прецедента	UC - 1
Актёры	ЕПГУ, Шлюз, ЕМИС, Витрина
Цель	Забронировать слот для консультации
Предусловия	1. Слот имеет статус AVAILABLE в таблице associations_slots. 2. У слота нет book_ext_id в таблице associations_slots.

Название прецедента	UC - 1
Основной сценарий	1. ЕПГУ отправляет запрос к Шлюзу на бронирование записи 2. Шлюз отправляет запрос в ЕМИС на бронирование записи 3. ЕМИС возвращает в ответе сгенерированный на своей стороне book_ext_id и статус-код бронирования. 4. Шлюз проверяет соответствие JSON-структуры ожидаемой в ответе (обязательные поля). Шлюз добавляет book_ext_id в таблицу associations_slots (таблица на стороне БД Шлюза). 5. Шлюз отправляет в Витрину слот со статусом RECORDED. 6. Шлюз отправляет в Витрину бирку со статусом 5000. 7. Шлюз изменяет статус слота на RECORDED в таблице associations_slots.

Входные данные объекта **Book**:

Атрибут	Комментарий
id	Используется поле book_ext_id из ответа на запрос бронирования.
patient_id	Используется поле patient_id из запроса бронирования
slot_id	Используется поле slot_id из запроса бронирования
status_code	По умолчанию 5000 (RECORDED) – пациент записан;
tag_source	По умолчанию EPGU
type	Используется booking_type из запроса на бронирование
referral_id	Используется referral_id из запроса на бронирование

6.2 Отмена бронирования выбранного слота

Метод POST /booking/cancel используется для отмены бронирования слота. ФЭР шлюз передает набор данных в ЕМИС / МИС для отмены бронирования, в ответном сообщении ЕМИС / МИС возвращает извещение о результатах обработки запроса.

Метод должен обрабатывать не более 10 секунд.

Общие сведения о методе и описание входных и выходных данных метода приведено в таблицах ниже.

Описание параметров в запросе:

№	Параметр	Тип данных	Обязательность	Описание
1	book_ext_id	string	О	Идентификатор записи
2	patient_id	string	О	Идентификатор пациента в РРП
3	slot_id	string	О	Идентификатор слота, на который было выполнено бронирование.
4	mo_oid	string	Н	Идентификатор медицинской организации (МО)
5	depart_oid	string	Н	Идентификатор подразделения
6	booking_source	string	Н	Источник записи INFOMAT

№	Параметр	Тип данных	Обязательность	Описание
				DOCTOR OTHER WEB REG

Пример запроса:

```
{
  "slot_id": "12345",
  "book_ext_id": "67890",
  "patient_id": "98765",
  "mo_oid": "43210",
  "depart_oid": "11223",
  "booking_source": "REG"
}
```

Ответ: Ответ на успешно выполненный запрос (200)

Пример ответа: Описание параметров в ответе:

№	Параметр	Тип данных	Обязательность	Описание
1	book_ext_id	string	O	Идентификатор записи на стороне РМИС Данный идентификатор необходимо использовать в качестве id записи при выгрузке бирки (Таблица 7 – Book - записи на бирку пациентов, поле – id)
2	patient_id	string	O	Идентификатор пациента GUID
3	slot_id	string	O	Идентификатор слота
4	mo_oid	string	H	Идентификатор медицинской организации (МО) OID
5	depart_oid	string	H	Идентификатор подразделения
6	booking_source	string	H	Источник записи. Возможные значения: INFOMAT, DOCTOR, OTHER_WEB

Пример успешного ответа:

```
{
  "slot_id": "12345",
  "book_ext_id": "67890",
  "type": "string",
  "status": {
    "statusCode": 200,
    "statusMessage": "Бронирование успешно отменено"
  }
}
```

Описание параметров в ответе для неуспешной отмены брони:

№	Параметр	Тип	Обязательность	Описание
---	----------	-----	----------------	----------

		данных		
2	status	object	O	Статус ответа
2.1	statusCode	integer	O	Код статуса
2.2	statusMessage	string	H	Информационное сообщение статуса (лучше отправлять)

Пример ответа с бизнес ошибкой:

```
{
  "book_ext_id": "597bc82e-88a5-4be5-8bd2-51a54312dbef",
  "status": {
    "status_code": 1,
    "status_message": "Запись с ИД записи уже отменена"
  }
}
```

6.2.1 Оптимизация метода отмены бронирования записи

Сценарий:

Название прецедента	UC - 1.1
Актёры	ЕПГУ, Шлюз, ЕМИС, Витрина
Цель	Отменить запись через ЕПГУ при бронировании через ЕПГУ
Предусловия	1. Слот имеет статус RECORDED в таблице associations_slots. 2. У слота есть book_ext_id в таблице associations_slots.
Основной сценарий	1. ЕПГУ отправляет запрос к Шлюзу на отмену бронирования записи 2. Шлюз отправляет запрос в ЕМИС на отмену бронирования записи 3. ЕМИС возвращает положительный ответ на отмену бронирования записи. Шлюз проверяет соответствие JSON-структуры ожидаемой в ответ (обязательные поля). 4. Шлюз удаляет book_ext_id из таблицы associations_slots. 5. Шлюз обновляет в Витрине слот со статусом AVAILABLE. 6. Шлюз обновляет в Витрине бирку со статусом 5010. 7. Шлюз изменяет статус слота на AVAILABLE в таблице associations_slots.

6.3 Коды ошибок

http-код	Текст ошибки	Описание ошибки
200	Запрос успешно обработан	
400	Ошибка в запросе, информация об ошибке содержится в параметре «ettoг»	
401	Отсутствует токен авторизации или передан не валидный токен	Описание в п.6.4
403	Нет полномочий на выполнение запроса	
404	Результат по заданному идентификатору SQL-запроса не найден	
405	Обновление информации о структурных подразделениях МО [***]	В случае обнаружения

http-код	Текст ошибки	Описание ошибки
	заблокировано. Для разъяснения обратитесь в группу сопровождения: «Витрина данных ХМАО».	большого количества технических ошибок со стороны МО, подразделение будет переведено в статус enabled = NONE (не принимает).
406	Неподдерживаемая версия протокола (после внедрения поддержки обратной совместимости возвращается только для несуществующих версий)	
422	Дублирование данных	Примеры дублей описаны в Раздел 5.1, 5.4.
500	Системная ошибка (в связи с принятыми ограничениями по доступности код 500 предполагается только при сбое Агента)	
503	Выполняются технические работы, отправка данных приостановлена до окончания работ.	

Коды ошибок в ответе на запрос бронирования слота:

Код ошибки status.statusCode	Описание ошибки	Отображаемый текст для пользователя
0		Переход на экран успешной записи – не отображается пользователю
11	Бирка к врачу занята или заблокирована Слот заблокирован другим пациентом Слот занят или заблокирован Слот заблокирован другим пациентом	Выбранное время занято Вы не успели подтвердить бронирование. Выберите другое время
12	Есть бирка на то же время	У вас есть запись к другому специалисту на то же время У вас уже есть запись к другому специалисту на то же время. Отмените ее или выберите другое время для записи
13	Пациент имеет предстоящую запись к данному врачу или врачу данной специальности Есть бирка на тот же день к тому же врачу	У вас есть активная запись к специалисту на тот же день Записаться снова можно после того, как эта запись станет недействительна. Вы можете отменить её или записаться на другой день
14	Бирка только для пациентов ДО ГОДА Бирка только для пациентов СТАРШЕ ГОДА Бирка только для детей до 18 лет Бирка только для пациентов от 18 лет	Слот не соответствует по возрасту Вы не можете записаться к этому специалисту, так как он ведет прием для другой возрастной категории пациентов
15	Слот аннулирован	Выбранное время недоступно Данный слот был аннулирован. Пожалуйста, выберите другое время
17	Пациент не найден	Пользователь не найден Проверьте правильность указанных данных в личном кабинете
18	Бирка не соответствующего типа Извините, запись невозможна	Внутренняя ошибка сервиса Ошибка при получении данных из медицин-

	Передано неверное направление. Слот не найден Несоответствие в переданных данных для записи Ошибка снятия блокировки	ской организации
--	--	------------------

Коды ошибок в ответе на запрос отмены бронирования:

Код ошибки status.statusCode	Описание ошибки	Отображаемый текст для пользователя
0		Переход на экран успешной записи – не отображается пользователю
1	Запись с таким ИД уже отменена	Ошибка при отмене записи Произошла ошибка. Попробуйте снова или зайдите позже. Если ничего не изменится, напишите в службу поддержки
18	Произошла техническая ошибка при отмене в МИС	Ошибка при отмене записи Произошла ошибка. Попробуйте снова или зайдите позже. Если ничего не изменится, напишите в службу поддержки

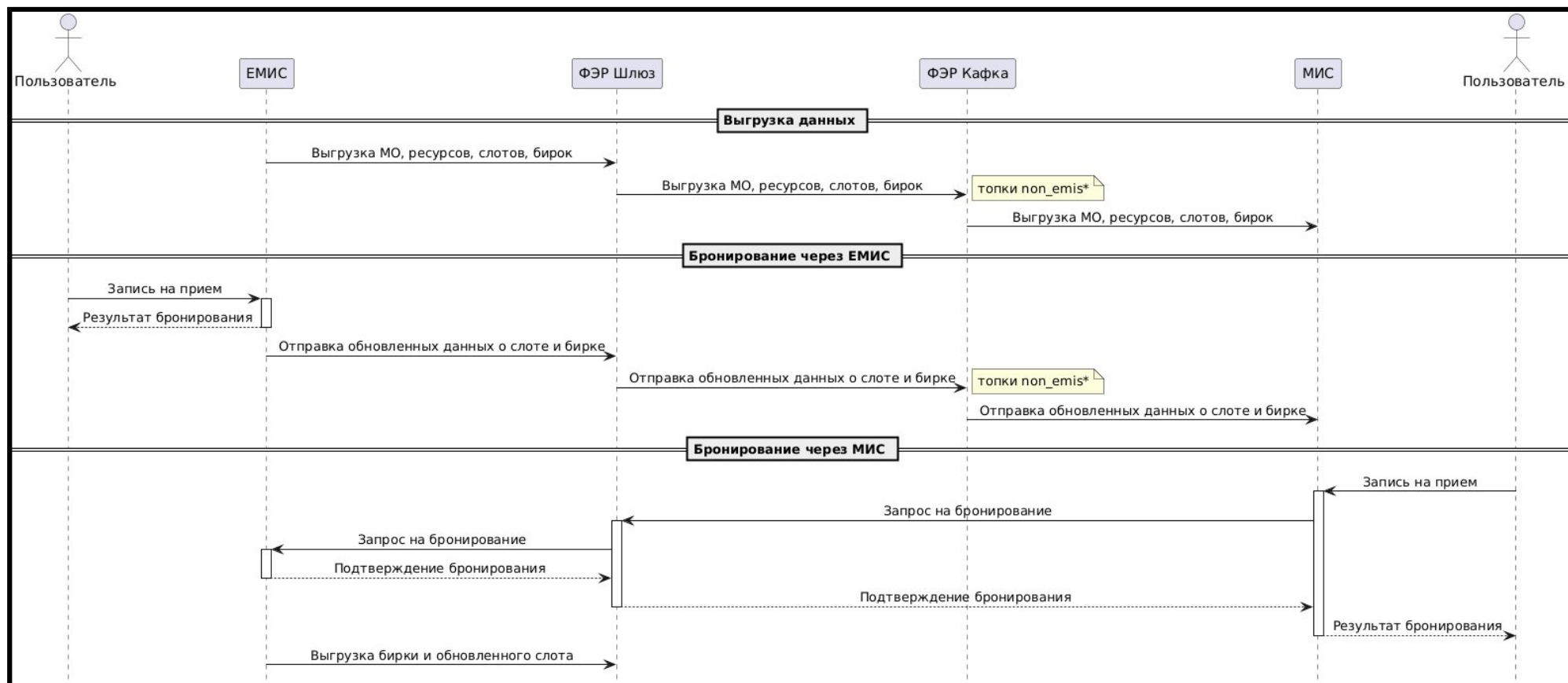


Рисунок 7 – Диаграмма последовательностей, при источнике подтверждения бронирования EMIS

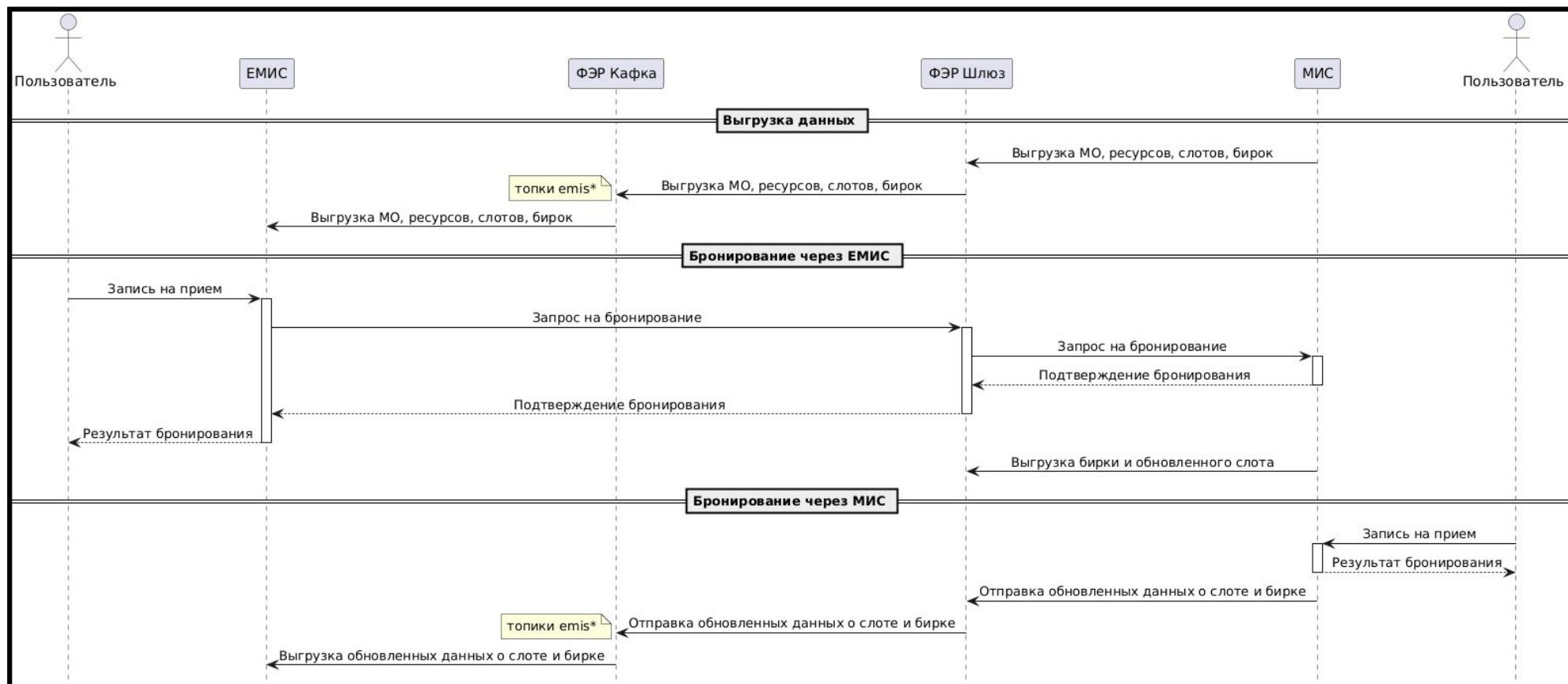


Рисунок 8 – Диаграмма последовательностей, при источнике подтверждения бронирования МИС

6.4 Авторизация при обращении к адаптеру бронирования

Все обращения к Адаптеру бронирования **обязаны выполняться с использованием авторизационного токена**, передаваемого в заголовке запроса **x-api-key**.

Используемый тип авторизации — **API KEY**, представляет собой уникальный идентификатор, применяемый для аутентификации и авторизации взаимодействия между системами.

6.3.1. Правила обработки запросов

При обработке запросов следует следующей логике:

1. Если заголовок **x-api-key** отсутствует — вернется ошибка HTTP 401 Unauthorized.
2. Если указан не валидный токен — вернется ошибка HTTP 401 Unauthorized.
3. Если передан валидный токен — вернется ответ в соответствии с запросом

Пример запроса:

```
curl --location '<url>/MIS/booking/book' \
--header 'Content-Type: application/json' \
--header 'x-api-key: <AUTH_TOKEN>' \
--data '{
  "booking_type": "string",
  "slot_id": "string",
  "patient_id": "string",
  "mo_oid": "string",
  "depart_oid": "string",
  "preliminary_reservation": true,
  "referral_id": "string",
  "cards_id": "string",
  "email": "string",
  "mobile_phone": "string",
  "booking_source": "string"
}'
```

Токены выдаются по заявке в СКИТ

7 Запись на прием по направлению

7.1 Краткое описание

Услуга обеспечивает возможность записи пациента на прием к врачу по ранее созданному направлению.

В рамках услуги «Запись на прием по направлению» обеспечиваются следующие функциональные возможности:

- запись пациента на прием к врачам по ранее созданным направлениям в МО;
- возможность выбора в качестве пациента себя или своего ребенка;
- просмотр списка имеющихся направлений;
- автоматическое определение региона получения услуги на основании сведений, содержащихся в личном кабинете пользователя, а также изменение региона пользователем вручную;
- автоматическое определение МО и специальности врача/услуги в соответствии со сведениями, содержащимися в направлении;
- выбор врача (кабинета, аппарата) для получения медицинской услуги, указанной в направлении, с учетом принятых в субъекте РФ правил предоставления первичной медико-санитарной амбулаторной помощи;
- выбор доступных для записи даты и времени приема;
- на основании заявки на запись создание записи в СП МО, определенную в направлении, или в доступное для записи СП МО головной МО, определенной в направлении;
- отображение в ленте ЕПГУ факта записи, включая статус записи;
- обновления статуса со стороны ГИСЗ субъекта РФ;
- отмена записи по инициативе пользователя ЕПГУ;
- отмена записи по инициативе МО;
- подача жалобы на отсутствие направлений или слотов времени

7.2 Клиентский путь «Запись на прием по направлению в медицинскую организацию»

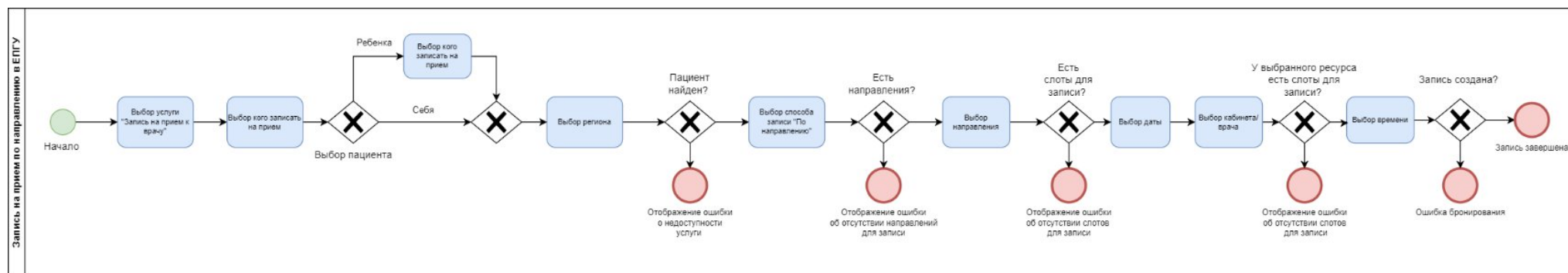


Рисунок 9 – Клиентский путь «Запись на прием по направлению в медицинскую организацию»

7.3 Сценарий «Запись на прием по направлению в медицинскую организацию»

Цель	Запись пациента на прием по направлению на ЕПГУ
Участники	1. Пользователь ЕПГУ (Пользователь) 2. ЕПГУ 3. Витрина данных 4. МИС 5. ФЭР Шлюз
Входная информация	Сведения о пациенте
Выходная информация	Сведения о записи на прием к врачу

Основной сценарий. Запись пациента на прием по направлению	
Предусловия	1. Пользователь авторизован на ЕПГУ 2. Пользователю доступна услуга «Запись на прием к врачу»

Последовательность (поток) действий	1. Пользователь выбрал услугу «Запись на прием к врачу» 2. Пользователь указывает пациента: себя; ребенка – в случае выбора «себя» ЕПГУ использует персональные данные Пользователя как пациента – в случае выбора «ребенка» Пользователь выбирает ребенка из списка зарегистрированных на ЕПГУ своих детей или вводит вручную персональные данные ребенка в составе: ФИО, СНИЛС, Дата рождения, пол, полис ОМС. ЕПГУ использует персональные данные ребенка как пациента; 3. Пользователь указывает регион оказания услуги 4. Пользователь указывает способ записи на прием «Запись по направлению» 5. ЕПГУ посредством ФЭР Шлюза отправляет запросы в Витрину и выполняет следующие действия: – получает список доступных для записи направлений пациента; – определяет наличие активных записей пациента 6. Пользователь выбирает направление 7. ЕПГУ посредством ФЭР Шлюза отправляет запрос в Витрину данных и получает информацию о доступных слотах 8. Пользователь выбирает дату записи 9. Пользователь выбирает ресурс (врача/аппарат/кабинет) 10. ЕПГУ посредством ФЭР Шлюза отправляет запрос в Витрину данных и получает список доступных слотов для ресурса, выбранного пользователем 11. Пользователь выбирает слот 12. ЕПГУ посредством ФЭР Шлюза передает в МИС информацию о запросе на создание записи на выбранный пользователем слот 13. МИС получает запрос на создание записи на слот, проверяет возможность создания записи и далее, в случае успеха – создает запись – передает в ФЭР Шлюз сведения об успешном создании записи
Результат (постусловия)	Создана запись на прием по направлению в медицинскую организацию

Альтернативный сценарий (Исключение) 1. Нет направлений для записи	
Последовательность (поток) действий	1. Выполняются шаги 5 основного сценария 2. ЕПГУ отображает ошибку о том, что нет направлений для записи
Альтернативный сценарий (Исключение) 2. Нет доступных слотов	
Последовательность (поток) действий	1. Выполняется шаг 7 основного сценария 2. ЕПГУ отображает ошибку о том, что нет доступного времени для записи
Альтернативный сценарий (Исключение) 3. У выбранного ресурса закончилось время	
Последовательность (поток) действий	1. Выполняется шаг 9 основного сценария 2. ЕПГУ отправляет запрос в Витрину данных и получает информацию об отсутствии слотов для выбранного пользователем врача 3. ЕПГУ отображает одну из ошибок: – запись к врачу на этот день закончилась – запись в кабинет на этот день закончилась
Альтернативный сценарий (Исключение) 4. Ошибка бронирования слота	
Последовательность (поток) действий	1. Выполняется шаг 12 основного сценария 2. МИС получает запрос на создание записи на слот, передает на ЕПГУ сведения о невозможности создать запись 3. ЕПГУ отображает одну из ошибок бронирования: – выбранное время занято – у вас есть запись к другому специалисту на то же время – у вас есть активная запись к специалисту на тот же день – слот не соответствует по возрасту

	– выбранное время недоступно – пользователь не найден – внутренняя ошибка сервиса
--	---

7.4 Сценарий «Отмена записи на прием по направлению в медицинскую организацию»

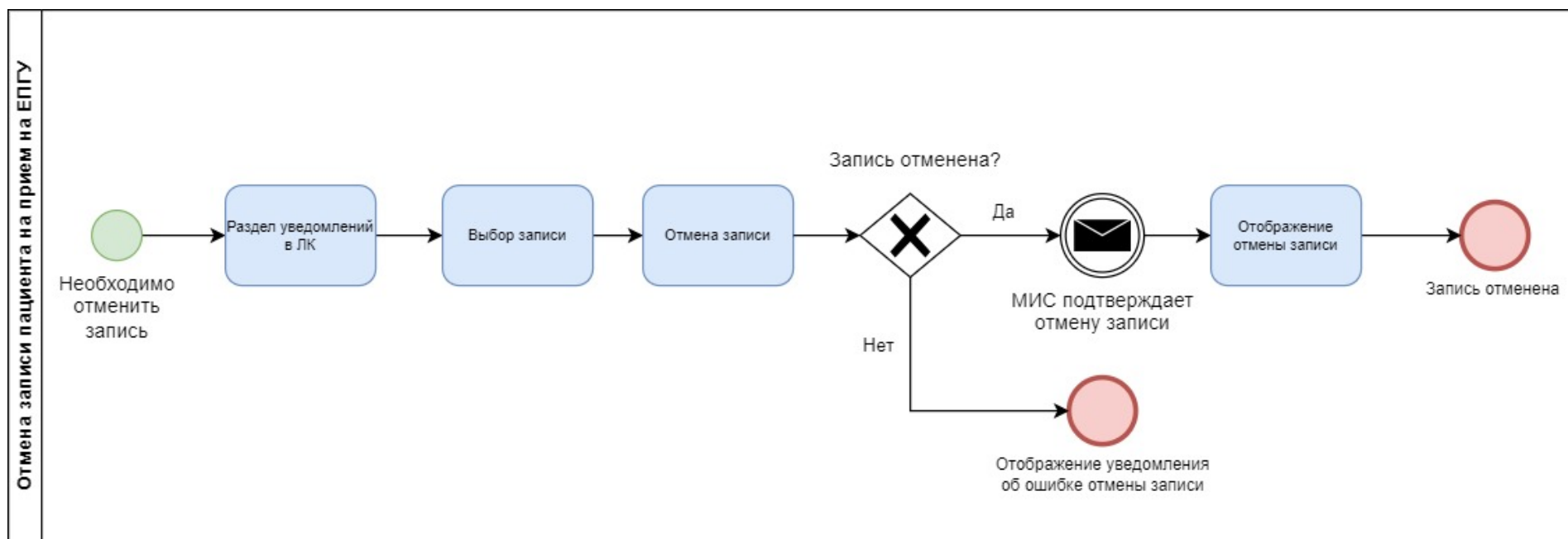


Рисунок 10 - Клиентский путь «Отмена записи пациента на прием по направлению в медицинскую организацию»

Основной сценарий. Отмена записи пациента на прием по направлению в медицинскую организацию	
Предусловия	1. Пользователь авторизован на ЕПГУ 2. Пользователю доступна услуга «Запись на прием к врачу»
Последовательность (поток) действий	1. Пользователь в личном кабинете или на финальном экране отменяет запись

	2. Пользователь подтверждает отмену записи 3. ЕПГУ посредством ФЭР Шлюза передает в МИС информацию о запросе на отмену записи 4. МИС получает запрос на отмену записи и далее – отменяет запись – передает на ЕПГУ сведения об отмене записи
Результат (постусловия)	Отменена запись на прием к врачу по направлению в медицинскую организацию

Альтернативный сценарий (Исключение) 1. Ошибка отмены записи на прием по направлению в медицинскую организацию	
Последовательность (поток) действий	1. Выполняется шаг 4 основного сценария – ЕПГУ получает сведения об ошибке при отмене записи 2. ЕПГУ отображает ошибку, что не удалось отменить запись на прием к врачу

7.5 Описание реализации услуги на ЕПГУ

ШАГ	Наименование экрана/шага	Описание экрана	Вызываемые методы/запросы (для получения данных, которые отображаются на экране)	Фильтры/Проверки	Действия Пользователя/системы	Результат
Шаг 6.2	Экран «Направление»	В списке направлений пользователю будут показаны все направления доступные ему в регионе	-	1) Для направлений в статусе "AVAILABLE" выводится кнопка "Записаться". 2) Для направлений в статусе "RECORDED" выводится информация о текущей записи из ответа Get_Books (маппинг по book_referral_id) и кнопка «Отменить запись»	Действия пользователя: клик на кнопку «Записаться» клик на кнопку «Отменить запись» Действия системы: переход на следующий шаг.	При клике на «Записаться» - переход на Шаг 7.2 При клике на «Отменить запись» - переход в сценарий отмены записи к врачу

				3) Направления в статусе "RECORDED", если нет записи в Get_Books по referral_id - не выводим (на направление уже была запись и эта запись в прошлом, считаем, что пользователь воспользовался)		
Шаг 7.2	Экран «Выберите дату и врача»	Отображается календарь с доступными днями для записи: - если есть слоты только через регистратуру – день отображается в синей рамке - если есть слоты для записи онлайн – день отображается синим - иначе (нет ни одного слота) – день не кликабельный	Входные параметры: если направление на <u>консультацию</u> - speciality_id – ИД должно-сти, из направления - mo_id - visittime_from - дата с - visittime_to - дата по (+ 14 дней) - profile_code - код профиля из направления если направление на <u>исследование</u> - service_code – код услуги из направления - mo_id - visittime_from - дата с - visittime_to - дата по (+ 14 дней) - profile_code - код профиля из направления Данные возвращаются сгруппированные по дням, ресурсам и по типу слота (по возрасту). Подсчитаны количество доступных/занятых сло-	1) Отфильтровать по slot_tag_age: если пациенту (кого записываем к врачу) ≥ 18 - берем записи slot_tag_age in ('ADULT', 'ALL') если пациенту < 18, но ≥ 1 год - берем записи slot_tag_age in ('CHILD', 'ALL') если пациенту меньше года - slot_tag_age in ('BABY', 'ALL'). 2) Проверить есть ли слоты на ближайшие 2 недели: $\text{sum(ergu)} > 0$ и $\text{sum(ergu)} \neq \text{sum(recorded_ergu)}$ - есть доступные для записи онлайн слоты $\text{sum(registry)} > 0$ и $\text{sum(registry)} \neq \text{sum(recorded_registry)}$ - есть доступные слоты для записи через регистратуру.	Действия пользователя: клик на дату Действие системы: отобразить под календарем список МО и ресурсов	Переход на шаг 7.2.1

			тов для ЕПГУ/Регистратуры/дучета.			
Шаг 7.2.1	Экран "Выберите дату и врача"	Под календарем для выбранного дня отображаются список МО и ресурсов		1) Группировка полученных данных по МО (mo_id) 2) Внутри МО группируем по resource_id	Действия пользователя: клик на выбранного врача Действие системы: переход на следующий шаг	Переход на шаг 7.2.2
Шаг 7.2.2	Экран "Выберите дату и врача"	Для выбранного ресурса отображается список доступных слотов, и кнопка «Записаться»	Входные параметры: resource_id – ИД ресурса visittime_from – время с visittime_to – время по mo_id – ИД СП МО, к которому ресурс относится	1) Фильтр по параметру slot_tag_age: если пациенту (кого записываем к врачу) больше или равно 18 - отобразить слоты с slot_tag_age in ('ADULT', 'ALL') если пациенту меньше 18, но больше 1 года - отобразить слоты с slot_tag_age in ('CHILD', 'ALL') если пациенту меньше года - slot_tag_age in ('BABY', 'ALL').	Действия пользователя: выбор слота и клик по «Записаться» Действие системы: переход на следующий шаг	Переход на шаг 8.2
Шаг 8.2	Бронирование слота	Лоадер на экране (ожидаем ответ от МИС)	Входные параметры: - slotId – ИД выбранного слота - patient_id – ИД пациента в витрине - booking_type=APPOINTMENT - preliminaryReservation = false - referral_id – ИД направления		Действия системы: переход на следующий шаг	при получении успешного ответа – переход на Шаг 9.2. иначе – отображается экран с модальной ошибкой, см. Error: Reference source not found.
Шаг 9.2	Финальный экран записи	Пользователь записан к врачу		Фильтров и проверок нет	Действия: При клике на "Добавить в календарь" - осуществляется ска-	Успешная запись пользователя

					чивание для добавления события в календарь (ics-файл) При клике на "В личный кабинет" - переход на просмотр деталей записи в Ленте уведомлений	
--	--	--	--	--	--	--

7.6 Требования к услуге

- Направления выписываются в СП МО.
- Через ЕПГУ реализовано отображение и возможность записаться по направлению на консультацию и на исследование:
 - Направление на консультацию – это направление к специалисту, ресурсом является врач (но не конкретный, а известна только специальность).
 - Направление на исследование – это услуга, ресурсом является кабинет или аппарат.
- Направление назначается по профилю медицинской помощи. Ресурс, которому назначаются слоты, должен относиться к коду профиля в направлении.

1. Данные тянутся из табл. referral, mo, service (если направление на исследование), resource, profilecode_resource, slot:

a. referral:

i. в зависимости от типа направления обязательно должны быть заполнены параметры направления:

- если type = 'CONSULTATION': id, patient_id, start_date, to_mo_id, to_profile_code, status, speciality_id, to_post_name;

- если type = 'RESEARCH': id, patient_id, start_date, to_mo_id, to_profile_code, status, to_service_id, to_service_name.

Остальные поля опционально, см. правило заполнения.

ii. поле status:

- направление, которое доступно для записи имеет статус AVAILABLE.
- направление, на которую есть активная запись, имеет статус RECORDED (есть запись в табл. book, с указанием referral.id)

b. **service:**

- поле code = код услуги (= значению to_service_id в направлении);
- resource_id = идентификатор ресурса, на которого выписаны слоты

c. **resource:**

- поле mo_id = идентификатор записи СП МО (mo.id), т.е. СП МО, к которому относится ресурс;
- признак enabled не равен 'NONE', т.е. ресурс должен быть «активный»;

d. **profilecode_resource:**

- resource_id = идентификатор ресурса, который относится к профилю назначения направления;
- profile_code = код профиля назначения направления (= значению to_profile_code в направлении);

e. **slot:**

- тип слота равен одному из значений (tag_type) 'REFERRAL', 'ORDINARY', 'EPGU', 'REGISTRY'
- слот доступен для записи (status='AVAILABLE')
- признак тип слота по возрасту (tag_age) должен быть заполнен в соответствии с правилами – по нему фильтр на фронте с учетом возраста пациента.

2. На экране выбора слота дополнительно осуществляется проверка доступности ресурса (период недоступности ресурса пересекается с интервалом времени, за который запрашиваются слоты, табл. *unaccessible_period*).
3. После успешного бронирования слота:
 - a. поле status в табл. slot принимает значение RECORDED.
 - b. в табл. book появляется новая запись с type = 'APPOINTMENT' со status_code = 5000.
 - c. в табл. book обязательно заполняется поле referral_id – равно значению id из табл. referral;**
 - d. поле status в табл. referral принимает значение RECORDED. Если пользователь отменяет запись по направлению – значение поля становится равно AVAILABLE (т.е. направление повторно доступно для записи).

8 Ответный сервис

Сервис предназначен для передачи в ЕМИС данных записи на прием к врачу. На адрес зарегистрированного вебхука будут отправляться запросы бронирования и отмены пользователем бронирования слота.

Перед регистрацией ответного сервиса, необходимо убедиться, что доступны запросы с адресов:

- 10.86.6.189
- 10.86.6.237

8.1 Регистрация вебхука

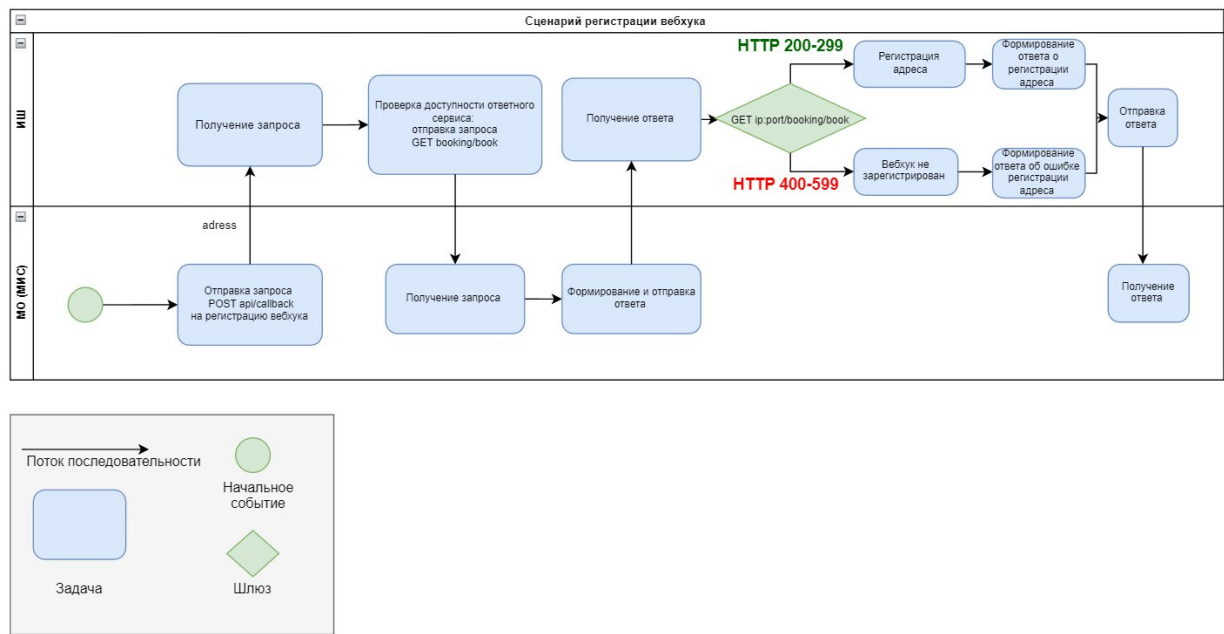


Рисунок 11 - Сценарий регистрации вебхука

Цель	Зарегистрировать адрес ЕМИС/МИС для получения запросов бронирования слота
Участники	1. ЕМИС 2. МИС 3. ФЭР Шлюз
Входная информация	URL адрес сервиса, на который будут поступать запросы бронирования
Выходная информация	Результат выполнения запроса
Последовательность (поток действий)	1. ЕМИС/МИС отправляет запрос на добавление ответного сервиса для бронирования слотов. 2. ФЭР шлюз принимает запрос от ЕМИС/МИС. 3. ФЭР шлюз проверяет доступность ответного сервиса, отправив запрос GET {ip:port}/booking/book. 4. Ответный сервис получает запрос. 5. Ответный сервис отправляет ответ 200 . 6. ФЭР шлюз получает ответ и регистрирует адрес ответного сервиса. 7. ФЭР шлюз отправляет успешный ответ на запрос о регистрации ответного

	сервиса.
Альтернативный поток 5а.	1. Ответный сервис отправляет ответ: 400/500/404/502. 2. ФЭР шлюз получает ответ. 3. ФЭР шлюз формирует и отправляет ответ об ошибке регистрации ответного сервиса.

Доступна регистрация одного вебхука.

Адрес	http://<адрес сервера>/api/callback
Метод	POST
Входные данные	address – URL адрес сервиса, на который будут поступать запросы бронирования
Выходные данные	Структура ответа описана в Таблица 2 - Структура записи о вебхуке

В случае, если необходимо зарегистрировать отдельный вебхук для структурного подразделения, необходимо указывать OID подразделения.

Адрес	http://<адрес сервера>/api/callback
Метод	POST
Входные данные	address – URL адрес сервиса, на который будут поступать запросы бронирования depart_oid – OID подразделения
Выходные данные	Структура ответа описана в Таблица 2 - Структура записи о вебхуке

8.2 Получение вебхука

Адрес	http:// <адрес сервера>/api/callback
Метод	GET
Выходные данные	Структура ответа описана в Таблица 2 - Структура записи о вебхуке

Получение вебхука подразделения:

Адрес	http:// <адрес сервера>/api/callback?depart_oid={depart_oid}
Метод	GET
Входные данные	depart_oid – OID подразделения
Выходные данные	Структура ответа описана в Таблица 2 - Структура записи о вебхуке

8.3 Обновление вебхука

Адрес	http:// <адрес сервера>/api/callback
Метод	PUT
Входные данные	address – URL адрес сервиса, на который будут поступать запросы бронирования
Выходные данные	Структура ответа описана в Таблица 2 - Структура записи о вебхуке

Обновление вебхука подразделения:

Адрес	http:// <адрес сервера>/api/callback
-------	--------------------------------------

Метод	PUT
Входные данные	address – URL адрес сервиса, на который будут поступать запросы бронирования depart_oid – OID подразделения
Выходные данные	Структура ответа описана в Таблица 2 - Структура записи о вебхуке

8.4 Удаление вебхука

Адрес	http:// <адрес сервера>/api/callback
Метод	DELETE

Удаление вебхука подразделения:

Адрес	http:// <адрес сервера>/api/callback?depart_oid={}
Метод	DELETE
Входные данные	depart_oid – OID подразделения
Выходные данные	Структура ответа описана в Таблица 2 - Структура записи о вебхуке

После удаления вебхука, МИС не будет получать запросы о бронировании.

Таблица 2 - Структура записи о вебхуке

Элемент	Кратность	Описание
address	1..1	Ссылка на адрес ответного сервиса. Данный адрес должен быть доступен с контура

8.5 Проверка доступности вебхука

Адрес	http:// <адрес сервера>/api/callback/check
Метод	POST
Входные данные	address - ссылка на адрес ответного сервиса
Выходные данные	Результат доступности ответного сервиса

Пример подстановки адреса для регистрации вебхука:

webhook = http://<ip>:<port>/{endpoint_mis} (при наличии)

Операция сервиса	Формируемый webhook endpoint
Бронирование	{webhook}/booking/book
Отмена	{webhook}/booking/cancel
Вызов врача на дом	{webhook}/homeVisit/create
Отмена вызова врача на дом	{webhook}/homeVisit/cancel

При регистрации вебхука указывать booking/book, booking/cancel, homeVisit/create, homeVisit/cancel не требуется. Сервис регистрации вебхука подставит окончание самостоятельно!

9 Адреса сервисов интеграционного шлюза

Таблица 3 - Адреса сервиса интеграционного шлюза ФЭР 2.0

Веб-сервис по приему данных от МО	
Название	Адрес
Тестовый контур	http://10.86.6.237:2100
Промышленный контур	http://10.86.6.189:2100

Адаптер бронирования	
Название	Адрес
Тестовый контур	http://10.86.6.237:2101
Промышленный контур	http://10.86.6.189:2101

Приложение 1. Структура сведений сервиса «Витрина данных»

Таблица 1 – МО - структурные подразделения

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Уникальный идентификатор структурного подразделения	id	id	O	string(45)	GUID Формат: нижний регистр, через дефис	Да
Уникальный идентификатор (OID) учреждения в системе здравоохранения	parent_id	parent_id	O	string(45)	Соответствует OID МО в справочнике Реестр медицинских организаций Российской Федерации 1.2.643.5.1.13.13.11.1461 Поле – OID	Да
Уникальный идентификатор подразделения	branch_oid	branch_id	O	string	Соответствует OID структурного подразделения в справочнике Реестр медицинских организаций Российской Федерации 1.2.643.5.1.13.13.99.2.114 Поле – OID структурного подразделения (depart_oid) Только подразделения с типом: 1 - Амбулаторный, 3 - Лабораторно-диагностический, 4 - Инструментально-диагностический	Нет

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Наименование структурного подразделения	name	name	О	string(512)		Да
Наименование подразделения медицинской организации	branch_name	branch_name	Н	string(512)	Полное наименование (OID) подразделения медицинской организации в ФРМО. Должно соответствовать значению в ФРМО.	
Наименование медицинской организации	parent_name	parent_name	О	string(512)	Полное наименование медицинской организации в ФРМО Значение должно соответствовать значению в ФРМО Заполнение поля обязательно для всех новых записей.	
Адрес структурного подразделения	address	address	О	string(512)		Да
Глобальный идентификатор адреса подразделения в ФИАС	address_fias_guid	address_fias_guid	Н	string(36)	Глобальный уникальный идентификатор адресного объекта ГАР ФИАС	Нет
ОКАТО региона	region_okato	region_okato	Н	string(11)		Нет

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Телефон подразделения	phone	phone	Н	string(45)		Да
Типы прикрепления, допустимые к данному подразделению	enabled	enabled	О	string	ATTACHED_MO - принимает по прикреплениям; ATTACHED_REG - принимает по прикреплению в регионе; ALL - принимает всех без прикрепления; NONE - не принимает	Да
Время создания записи	create_ts	create_ts	О	string	Формат: YYYY-MM-DDTHH:MI:SS ФОРМАТ RFC3339 (ISO8601)	Нет
Время обновления записи	update_ts	update_ts	О	string	Формат: YYYY-MM-DDTHH:MI:SS ФОРМАТ RFC3339 (ISO8601)	Нет
Тип обслуживаемого населения по возрастному признаку	tag_age	tag_age	О	string (45)	Возможные значения: ADULT – взрослый (с 18 лет); CHILD – детский (от 0 до 17 лет вкл.); ALL – все возрастные группы	Да

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Лицензия на осуществление соответствующих видов деятельности	license	license	Н	string(512)		Нет
Адрес электронной почты	email	email	Н	string(512)		Нет
Идентификатор ТВСП	territorial_department_id	territorial_department_id	Н	string(512)	Идентификатор территориально-выделенного структурного подразделения (ТВСП). Указывается в случае, если услуга оказывается в здании (по адресу) размещения ТВСП. Должен соответствовать значению в таблице ТВСП territorialDepart в структуре данных ФРМО (см. атрибут moTerritorialDepartmentId)	Нет
Наименование ТВСП	territorial_department_name	territorial_department_name	Н	string(512)	Наименование территориально-выделенного структурного подразделения Указывается в случае, если услуга оказывается в здании (по адресу) размещения ТВСП Должен соответствовать значению в таблице ТВСП territorialDepart в структуре данных ФРМО (см. атрибут moTerritorialDepartmentName)	Нет

Таблица 2 – Resource - ресурсы МО

Наименование	Атрибут	Имя атрибута в то- пике Kafka	Обязатель- ность	Тип данных	Комментарий	Влияние на отоб- ражение формы услуги
Общие параметры						

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Уникальный идентификатор ресурса	id	id	O	string (45)	GUID Формат: нижний регистр, через дефис	Да
Идентификатор записи личного дела ресурса	frmr_card_id	frmr_card_id	H	string (45)	Идентификатор записи для личного дела врача. Формат: GUID в нижнем регистре, через дефис. Значение идентификатора соответствует значению в ФРМР (поле person/card.id).	Нет
Тип ресурса	type	type	O	string	SPECIALIST – врач; MACHINE - медицинский аппарат; ROOM - кабинет	Да
Уникальный идентификатор структурного подразделения	mo_id	mo_id	O	string(45)	GUID Формат: нижний регистр, через дефис	Да
Типы прикрепления, допустимые к данному ресурсу	enabled	enabled	O	string	ATTACHED_MO - принимает по прикреплениям ATTACHED_REG - принимает по прикреплению в регионе ALL - принимает всех без прикрепления NONE - не принимает	Да
Время создания записи	create_ts	create_ts	O	string	Формат: YYYY-MM-DDTHH:MI:SS ФОРМАТ RFC3339 (ISO8601)	Нет
Время обновления записи	update_ts	update_ts	H	string	Данное поле обновляется на стороне Сервиса, следует прекратить отправку.	Нет
Дата рождения	birth_date		H	string(512)	Формат: YYYY-MM-DDTHH:MI:SS ФОРМАТ RFC3339 (ISO8601)	Нет
Специалист						
Код должности	post_id	post_id	O	string(45)	Код должности согласно справочнику НСИ 1.2.643.5.1.13.13.11.1002. Перечень кодов должностей мед. работников:	Да

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
					110 - врач-терапевт участковый; 59 - врач-педиатр участковый; 49 - врач общей практики (семейный врач); 122 - врач-хирург; 13 - врач-акушер-гинеколог; 54 - врач-офтальмолог; 53 - врач-оториноларинголог; 100 - врач-стоматолог; 103 - врач-стоматолог-терапевт; 101 - врач-стоматолог детский; 83 - врач-психиатр детский; 119 - врач-фтизиатр; 87 – врач психиатр-нарколог; 28 - врач-детский хирург; 120 – врач-фтизиатр участковый.	
Идентификатор специальности	speciality_id	speciality_id	O	string(45)	Обязательно при type = SPECIALIST, при type != SPECIALIST – не указывается) Значение Идентификатора специальности соответствует значению из справочника ФНСИ «Номенклатура специальностей специалистов со средним, высшим и послевузовским медицинским и фармацевтическим образованием в сфере здравоохранения» (OID 1.2.643.5.1.13.13.11.1066)	Да
Врач, СНИЛС	snils	snils	O	string(11)	Обязательно при type = SPECIALIST, при type != SPECIALIST – не указывается)	Нет
Врач, фамилия	last_name	lastname	O	string(512)	Обязательно при type = SPECIALIST, при type != SPECIALIST – не указывается)	Да

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Врач, имя	first_name	firstname	О	string(512)	Обязательно при type = SPECIALIST, при type != SPECIALIST – не указывается)	Да
Врач, отчество	middle_name	middlename	Н	string(512)		Да
Описание	description	description	Н	string(512)		Нет
Дата начала работы	begin_date		Н	string(512)	Формат: YYYY-MM-DDTHH:MI:SS ФОРМАТ RFC3339 (ISO8601)	Нет
Аппарат						
Аппарат, наименование	machine_name	machine_name	У	string(512)	Обязательно при type = MACHINE, при type != MACHINE – не указывается)	Да
Серийный номер оборудования	serial_number	serial_number	У	string	Обязательно при type = MACHINE, при type != MACHINE – не указывается)	Да
Служба						
Кабинет, номер	room_number	room_number	О	string(45)	Обязательно при type = ROOM, при type != ROOM – не указывается)	Да
Кабинет, наименование	room_name	room_name	О	string(512)	Обязательно при type = ROOM, при type != ROOM – не указывается)	Да

Таблица 3 – Service – услуги

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Уникальный идентификатор услуги	id	id	О	string(45)	Уникальный идентификатор, не привязанный ни к какой существующей таблице (ранее тут использовалось значение из таблицы связей Службы и Услуги) GUID	Нет

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
					Формат: нижний регистр, через дефис	
Идентификатор ресурса	resource_id	resource_id	О	string(45)	GUID Формат: нижний регистр, через дефис	Да
Наименование медицинской услуги	name	name	Н	string(512)	Для услуг по медицинскому освидетельствованию заполняется по данным справочника услуг МО; для остальных: - Заполняется согласно справочнику ФНСИ - Значение наименования услуги соответствует значению из справочника ФНСИ «Номенклатура медицинских услуг» (1.2.643.5.1.13.13.11.1070)	Да
Код услуги	code	code	О	string(45)	Код услуги согласно справочнику НСИ 1.2.643.5.1.13.13.11.1070 "Номенклатура медицинских услуг"	Да
Цена услуги	price	price	Н	number		Нет
Время создания записи	create_ts	create_ts	О	string	Формат: YYYY-MM-DDTHH:MI:SS ФОРМАТ RFC3339 (ISO8601)	Нет
Время обновления записи	update_ts	update_ts	Н	string	Формат: YYYY-MM-DDTHH:MI:SS ФОРМАТ RFC3339 (ISO8601)	Нет
Дата открытия	date_beg	не передается	Н	string(45)	Эти данные нужны для: 1. Определения актуальности услуги. Услуга, имеющая дату закрытия, признак и дату удаления считается не актуальной. 2. Разбора проблемных ситуаций в будущем.	Нет
Дата закрытия	date_end	не передается	Н	string(45)		Нет
Признак удаления	is_deleted	не передается	Н	boolean		Нет
ОИД справочника НСИ	dictionary_nsi_oid	не передается	Н	string(45)	Для вакцинации: dictionary_nsi_oid = 1.2.643.5.1.13.13.99.2.1074 Для остальных услуг (в случае записи на услугу): 1.2.643.5.1.13.13.99.2.1099	Нет

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Код записи из справочника НСИ	dictionary_nsi_code	не передается	Н	string(45)	Код услуги. Для вакцинации значение берётся из справочника 1.2.643.5.1.13.13.99.2.1074 Например: «9» Для остальных услуг значение берётся из справочника 1.2.643.5.1.13.13.99.2.1099. Например: «11»	Нет

Таблица 4 – Slot - слоты

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Уникальный идентификатор слота	id	id	О	string(45)	GUID Формат: нижний регистр, через дефис	Да
Уникальный идентификатор ресурса	resource_id	resource_id	О	string(45)	GUID Формат: нижний регистр, через дефис	Да
Типы записей	tag_type	tag_type	О	string	EPGU - для ЕПГУ; ORDINARY – обычный; RESERVE – резервный; PAID – платный; CENTER – для центра записи; REFERRAL – по направлению; D_OBSERVATION – для записи по карте диспансерного наблюдения; INFOMAT – для информата; REGISTRY – для регистратуры EQUEUE – для живой очереди; ONLINE – для приема средствами видеосвязи	Да

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
					DOCTOR - для записи типа врач-врач (недоступен для пользователя на ЕПГУ) У одного слота может быть только один тип. Если слот может быть доступен в нескольких сценариях, необходимо установить тип ORDINARY. Для ЕМИС: DISPENSARY - диспансеризация взрослого населения; PROPHYLACTIC - профилактические медицинские осмотры; DISPENSARY_DEEP - углубленная диспансеризация взрослого населения; VACCINATION - вакцинация; EMERGENCY - неотложный прием; AT_HOME - посещение на дому;	
Учет возраста	tag_age	tag_age	O	string	ADULT – взрослый, с 18 лет; CHILD – детский, с 1 года до 18 лет; BABY – младенец, от 0 до 1 года; ALL – все возрастные группы YOUNG – дети любого возраста (от 0 мес. до 17 лет вкл.)	Да
Время начала приема	visit_time	visittime	O	string	Формат: YYYY-MM-DDTHH:MI:SS	Да
Продолжительность в минутах процедуры оказания государственной услуги по регламенту	duration	duration	O	integer		Да

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Статус слота	status	status	О	string	AVAILABLE – доступен; RECORDED – записан; CANCELED – отменен; RESERVED – забронирован; BLOCKED - заблокирован	Да
Дата создания записи	create_ts	create_ts	О	string	Формат: YYYY-MM-DDTHH:MI:SS FORMAT RFC3339 (ISO8601)	Нет
Дата обновления записи	update_ts	update_ts	Н	string	Формат: YYYY-MM-DDTHH:MI:SS FORMAT RFC3339 (ISO8601)	Нет
Кабинет	cabinet	cabinet	Н	string		Нет
Комментарий к слоту	description	description	Н	string(512)		Нет
Ограничение по отделениям	restricted_sub_division	restricted_sub_division	Н	array [string]	Массив OID подразделений, которым будет доступна запись на данный слот. Соответствует OID структурного подразделения в справочнике Реестр медицинских организаций Российской Федерации 1.2.643.5.1.13.13.99.2.114	Нет
Ограничение по специальностям	restricted_speciality	restricted_speciality	Н	array [string]	Массив специальностей, которым будет доступна запись на данный слот. Коды должностей соответствуют справочнику 1.2.643.5.1.13.13.11.1002.	Нет
Квоты головных МО	mo_quota	mo_quota	Н	array [string]	Массив OID головных МО, для которых доступен котируемый слот. Поле передается в Kafka.	Нет

Таблица 7 – Book - записи на бирку пациентов

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Уникальный идентификатор записи на бирку	id	id	О	string	GUID Формат: нижний регистр, через дефис	Да

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
пациентов					При бронировании через ЕПГУ, данный идентификатор должен совпадать с book_ext_id.	
Уникальный идентификатор пациента	patient_id	patient_id	O	string(45)	GUID Формат: нижний регистр, через дефис	Да
Уникальный идентификатор слота	slot_id	slot_id	O	string(45)	GUID Формат: нижний регистр, через дефис	Да
Статус записи, код	status_code	status_code	O	integer	5000 (RECORDED) – пациент записан; 5010 (CANCELED) – запись отменена; 5020 (SUCCESS) – услуга оказана; 5030 (PATIENT_NOT_ARRIVED) – пациент не явился; 5040 (PATIENT_CANCELED) – запись отменена пациентом в МО	Да
Источник записи	tag_source	tag_source	O	string	EPGU – ЕПГУ; INFOMAT – инфомат; DOCTOR – врач; REG – регистратура; OTHER_WEB – другой источник записи CALL – запись через модуль Контакт-центр	Да
Тип записи	type	type	O	string	APPOINTMENT – прием к врачу; VACCINATION – вакцинация; COVID_VACCINATION – вакцинация от COVID-19; UDVN_DISPENSARY – усиленная диспансеризация; DISPENSARY – диспансеризация; PROPHYLACTIC – профилактический осмотр; D_OBSERVATION – запись по карте диспансерному наблюдению; ONLINE_CONSULTING – консультация онлайн;	Да

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
					PAID_APPOINTMENT – запись на платную услуги	
Идентификатор направления	referral_id	referral_id	O	string(45)	GUID Формат: нижний регистр, через дефис	Да
Дата создание записи	create_ts	create_ts	O	string	Формат: YYYY-MM-DDTHH:MI:SS ФОРМАТ RFC3339 (ISO8601)	Нет
Дата изменения записи	update_ts	update_ts	H	string	Формат: YYYY-MM-DDTHH:MI:SS ФОРМАТ RFC3339 (ISO8601)	Нет
Код услуги	service_code	не передается	H	string	Для услуги вакцинация значение service_code должно соответствовать dictionary_nsi_code в таблице service. Обязательно для записи на услугу вакцинация	Нет
Идентификатор услуги	service_id	не передается	Y	string	Ссылка на id записи в таблице service. Обязательно для записи на услугу вакцинация	Нет
Оригинальное происхождение записи пациента на слот	origin_source	не передается	Y	string	Инструкции для интеграции описана в Приложении	Нет
Источник последнего изменения записи пациента на слот	last_change_source	не передается	Y	string	Инструкции для интеграции описана в Приложении	Нет

Таблица 8 – Referral - направления пациентов

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Уникальный идентификатор направления пациента	id	id	O	string(45)	GUID Формат: нижний регистр, через дефис	Да
Идентификатор пациента	patient_id	patient_id	O	string(45)		Да
Идентификатор специаль-	speciality_id	speciality_id	Y	string(45)	Должно соответствовать коду из справочника ФНСИ	Да

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
ности					«Номенклатура специальностей специалистов со средним, высшим и послевузовским медицинским и фармацевтическим образованием в сфере здравоохранения» (OID 1.2.643.5.1.13.13.11.1066) Поле – код. ! Обязательно для направлений с type = CONSULTATION.	
Номер направления	number	number	O	string(45)		Да
Тип направления	type	type	O	string	CONSULTATION – на консультацию; EXAMINATION – на обследование; REHABILITATION – на восстановительное лечение; HOSPITALIZATION – на осмотр с целью госпитализации; RESEARCH – на исследование (используется на ЕПГУ); CONSULTING_ROOM – в консультационный кабинет; OUTPATIENT_VISIT – на поликлинический прием PROF_EXAMINATION – на проф. Осмотр VACCINATION – на осмотр перед вакцинацией VACCINATION_ROOM – в кабинет вакцинации TELEMED - онлайн-консультация (телемедицина). Необходимо обеспечить соответствие записи значению, указываемому в referral_type_code. Заполнение поля обязательно для всех новых записей	Да
Дата начала действия направления	start_date	start_date	O	string	Формат: YYYY-MM-DDTHH:MI:SS FORMAT RFC3339 (ISO8601)	
Дата окончания действия направления	end_date	end_date	H	string	Формат: YYYY-MM-DDTHH:MI:SS FORMAT RFC3339 (ISO8601)	

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
					Если направление бессрочное, может быть не заполнено	
Идентификатор структурного подразделения МО, в которое выдано направление	to_mo_id	to_mo_id	Н	string(45)	GUID СП МО Формат: нижний регистр, через дефис Если заполнено, должно относиться к головной МО из to_mo_oid; поиск слотов ограничивается этим СП МО. Если не заполнено, поиск выполняется по всем СП МО головной МО через to_mo_oid.	
Код профиля отделения куда выписано направление	to_profile_code	to_profile_code	У	string(45)	Значение должно соответствовать коду профиля из справочника ФНСИ "Профили медицинской помощи" (1.2.643.5.1.13.13.11.1119) Примечание: Необходимо выгрузить профиль ресурса с данным кодом, иначе расписание будет недоступно. Обязательно для type = CONSULTATION и type = RESEARCH.	
Наименование кода профиля медицинской помощи куда выписано направление	to_profile_name		Н	string(45)	Значение должно соответствовать наименованию профиля из справочника ФНСИ "Профили медицинской помощи" (1.2.643.5.1.13.13.11.1119)	Да
Наименование услуги	to_service_name		О	string(45)		Да
Наименование должности	to_post_name	to_post_name	У	string(512)	1.2.643.5.1.13.13.11.1002 "Должности медицинских и фармацевтических работников" !Обязательно для направлений с type = CONSULTATION.	Да
Код медицинской услуги, на которую выдано направление	to_service_id	to_service_id	О	string(45)	Код услуги согласно справочнику НСИ 1.2.643.5.1.13.13.11.1070 "Номенклатура медицинских услуг"	

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
					Примечание: Ссылается на запись в таблице Service, поле Code.	
Идентификатор структурного подразделения МО, выдавшее направление	from_mo_id	from_mo_id	Н	string(45)	GUID Формат: нижний регистр, через дефис	
Идентификатор ресурса, выдавшего направление	from_resource_id	from_resource_id	О	string(45)	GUID Формат: нижний регистр, через дефис	
Статус направления	status	status	О	string	AVAILABLE – доступно; RECORDED – записано; CANCELED – отменено;	
Дата создание записи	create_ts	create_ts	О	string	Формат: YYYY-MM-DDTHH:MI:SS FORMAT RFC3339 (ISO8601)	
Дата обновления записи	update_ts	update_ts	Н	string	Формат: YYYY-MM-DDTHH:MI:SS FORMAT RFC3339 (ISO8601)	
Идентификатор ресурса, к которому выдано направление	to_resource_id	to_resource_id	Н	string(45)	GUID Формат: нижний регистр, через дефис	
Идентификатор направления в МО ФОИВ	from_referral_id	from_referral_id	Н		Актуально только для направлений, создаваемых на Витрине расписаний МО ФОИВ	Нет
Идентификатор МИС создавшей направление в МО ФОИВ	from_rmis_id	from_rmis_id	Н		Актуально только для направлений, создаваемых на Витрине расписаний МО ФОИВ	Нет
Код медицинского направления	referral_type_code	referral_type_code	О		Значение должно соответствовать коду медицинского направления из справочника ФНСИ "Виды медицинских направлений" (1.2.643.5.1.13.13.11.1009)	Да
Название медицинского направления	referral_type_name	referral_type_name	О	string(512)	Значение должно соответствовать наименованию медицинского направления из справочника ФНСИ "Виды медицинских направлений" (1.2.643.5.1.13.13.11.1009)	Да

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Код диагноза	desease_code	desease_code	O	string(512)	Значение кода диагноза из справочника диагнозов МКБ-10, ФНСИ 1.2.643.5.1.13.13.99.2.568 Может быть заполнено несколькими значениями через запятую, например "J10.8, J11.0"	
OID учреждения, в которое выдано направление	to_mo_oid	to_mo_oid	O	string(45)	Значение уник. идентификатора МО (OID), значение ОИД головной организации. Должно соответствовать значению в ФРМО	
OID учреждения, выдавшего направление	from_mo_oid	from_mo_oid	O	string(45)	Значение уник. идентификатора МО или СП МО (OID). Значение уник. идентификатора МО или СП МО. Должно соответствовать значению в ФРМО	

Таблица 9 – Unaccessible_period - периоды недоступности ресурсов

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Уникальный идентификатор периода недоступности ресурса	period_id	period_id	O	string	GUID Формат: нижний регистр, через дефис	Да
Идентификатор ресурса	resource_id	resource_id	O	string(45)	GUID Формат: нижний регистр, через дефис	Да
Причина	reason	reason	O	string	VACATION – в отпуске; SICK_LEAVE – на больничном; REPAIR –на ремонте; ASSIGNMENT - в командировке; OTHER - прочие причины	Да
Начало периода	beg_date	beg_date	O	string	Формат: YYYY-MM-DDTHH:MI:SS ФОРМАТ RFC3339 (ISO8601)	Да

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Конец периода	end_date	end_date	O	string	Формат: YYYY-MM-DDTHH:MI:SS ФОРМАТ RFC3339 (ISO8601)	Да
Дата создание записи	create_ts	create_ts	O	string	Формат: YYYY-MM-DDTHH:MI:SS ФОРМАТ RFC3339 (ISO8601)	Нет
Дата изменения записи	update_ts	update_ts	H	string	Данное поле обновляется на стороне Сервиса, следует прекратить отправку.	Нет

Таблица 10 – Profilecode_resource

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Идентификатор записи профиля	id	id	O	string	GUID Формат: нижний регистр, через дефис	Да
Идентификатор ресурса	resource_id	resource_id	O	string(45)	GUID Формат: нижний регистр, через дефис	Да
Код профиля отделения, к которому привязан ресурс	profile_code	profile_code	O	string(45)	Значение должно соответствовать коду профиля из справочника ФНСИ "Профили медицинской помощи" (1.2.643.5.1.13.13.11.1119)	Да
Наименование профиля	profile_name	profile_name	O	string(45)	Значение должно соответствовать коду профиля из справочника ФНСИ "Профили медицинской помощи" (1.2.643.5.1.13.13.11.1119)	Да
Дата создание записи	create_ts	create_ts	O	string	Формат: YYYY-MM-DDTHH:MI:SS ФОРМАТ RFC3339 (ISO8601)	Нет
Дата создание записи	update_ts	update_ts	O	string	Формат: YYYY-MM-DDTHH:MI:SS ФОРМАТ RFC3339 (ISO8601)	Нет

Таблица 11 – Waiting_list - Лист ожидания

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Идентификатор листа ожидания	id	id	O	string	GUID Формат: нижний регистр, через дефис	Нет
Идентификатор ресурсов	resource_id	resource_id	H	array [string]	Массив GUID Только type = SPECIALIST	
Наименование листа ожидания	name	name	O	string(255)	Уникальное в пределах одной медицинской организации	
Идентификатор медицинской организации	mo_oid	mo_oid	O	string	Соответствует OID МО в справочнике Реестр медицинских организаций Российской Федерации 1.2.643.5.1.13.13.11.1461 Поле – OID	
Идентификатор специальности	speciality_id	speciality_id	O	string	Значение Идентификатора специальности соответствует значению из справочника ФНСИ «Номенклатура специальностей специалистов со средним, высшим и послевузовским медицинским и фармацевтическим образованием в сфере здравоохранения» (OID 1.2.643.5.1.13.13.11.1066)	
Идентификатор должности	position_id	position_id	O	string	Фед. справочник 1.2.643.5.1.13.13.99.2.181 ФРМР. Должности медицинского персонала (nsi."1.2.643.5.1.13.13.99.2.181".nsi_value_id)	
Учет возраста	tag_age	tag_age	H	string	ADULT – взрослый; CHILD – детский; BABY – младенец; YOUNG – Дети от 0 до 18 лет; ALL – все возрастные группы.	
Идентификатор подразделения	branch_oid	branch_oid	H	array [string]	Соответствует OID структурного подразделения в справочнике Реестр медицинских организаций Российской Федерации 1.2.643.5.1.13.13.99.2.114 Поле – OID структурного подразделения (depart_oid)	
Дата создания записи	create_ts	create_ts	O	string	Формат: YYYY-MM-DDTHH:MI:SS ФОРМАТ RFC3339 (ISO8601)	

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Дата изменения записи	update_ts	update_ts	Н	string	Формат: YYYY-MM-DDTHH:MI:SS FORMAT RFC3339 (ISO8601)	

Таблица 12 – Waiting_list_entity - Запись в листе ожидания

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Уникальный идентификатор листа ожидания	id	id	О	string(45)	GUID Формат: нижний регистр, через дефис	Нет
Идентификатор листа ожидания	waiting_list_id	waiting_list_id	О	string(45)	GUID Формат: нижний регистр, через дефис	
Идентификатор пациента в РРП	patient_id	patient_id	О	string(45)	GUID Формат: нижний регистр, через дефис	
Статус заявки в листе ожидания	status	status	О	string	WAITING - В ожидании; RECORDED - Переведена в запись; CANCELED - Исключен.	
Источник записи	source_record	source_record	Н	string	Значения для источника записи: CALL_CENTER - Региональный колл-центр (единственный номер телефона 122) OTHER_WEB – другой источник записи CONTACT_CENTER - Модуль "Контакт-центр" RECEPTION - Модуль "Регистратура"	
Дата создания записи	create_ts	create_ts	О	string	Формат: YYYY-MM-DDTHH:MI:SS FORMAT RFC3339 (ISO8601)	
Дата изменения записи	update_ts	update_ts	Н	string	Формат: YYYY-MM-DDTHH:MI:SS FORMAT RFC3339 (ISO8601)	

Таблица 13 – Home Visit – Вызовы врачей на дом

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Уникальный идентификатор записи	id	id	О	string(45)	GUID Формат: нижний регистр, через дефис	Да
Уникальный идентификатор пациента	patient_id	patient_id	О	string(45)	GUID Формат: нижний регистр, через дефис	Да
Уникальный идентификатор структурного подразделения	mo_id	mo_id	Н	string(45)	Соответствует OID МО в справочнике Реестр медицинских организаций Российской Федерации 1.2.643.5.1.13.13.11.1461 Поле – OID Уникальный идентификатор структурного подразделения id из таблицы mo. Условно-обязательное поле, т.к. должна быть возможность создания вызова врача на дом даже при невозможности автоматического определения МО вызова. Такие заявки будут отображаться в АРМ оператора Call-центра	Да
Наименование головной МО по ФРМО	mo_parent_name	mo_parent_name	Н	string	Условно-обязательное, может быть заполнено вместо mo_id, если не определено СП МО, но определен головной МО	Да
Идентификатор головной МО по ФРМО	mo_parent_id	mo_parent_id	Н	string		Да

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги																		
Статус записи, код	status_code	status_code	О	string	Возможные значения: 5000 (RECORDED) – Вызов на дом подтвержден; 5010 (CANCELED) – Вызов отменен МО; 5020 (SUCCESS) – Вызов выполнен, услуга оказана; 5030 (NOT_ACCOMPLISHED) – Вызов не состоялся по другим причинам; 5040 (PATIENT_CANCELED) – Вызов отменен пациентом; 5050 (REGISTERED) – Вызов зарегистрирован, ожидает подтверждения; 5060 (UNKNOWN_REGISTERED) – Вызов зарегистрирован, ожидает подтверждения, но МО не определено	Да																		
Идентификатор причины отказа/отмены вызова врача на дом	reason_id	reason_id	Н	string	Обязательно к заполнению при значениях статуса вызова врача на дом: 5030 NOT_ACCOMPLISHED 5010 CANCELLED 5040 PATIENT_CANCEL	Да																		
Причина отмены записи	reason_code	reason_code	Н	string	<table><tr><th>id</th><th>Наименование</th><th>Код</th></tr><tr><td>1</td><td>Необходимость вызова СМП</td><td>NEED_CALL_AMBULANCE</td></tr><tr><td>2</td><td>Необходимость вызова неотложной помощи</td><td>NEED_CALL_EMERGENCY_MEDICAL_CARE</td></tr><tr><td>3</td><td>Пациент явился на прием в МО</td><td>PATIENT_CAME_TO_THE_APPOINTMENT</td></tr><tr><td>4</td><td>Пациент ранее обслужен</td><td>HOME_VISIT_HAS_BEEN_SERVED</td></tr><tr><td>5</td><td>Дублирование</td><td>HOME_VISIT_DUPLICATE</td></tr></table>	id	Наименование	Код	1	Необходимость вызова СМП	NEED_CALL_AMBULANCE	2	Необходимость вызова неотложной помощи	NEED_CALL_EMERGENCY_MEDICAL_CARE	3	Пациент явился на прием в МО	PATIENT_CAME_TO_THE_APPOINTMENT	4	Пациент ранее обслужен	HOME_VISIT_HAS_BEEN_SERVED	5	Дублирование	HOME_VISIT_DUPLICATE	Да
id	Наименование	Код																						
1	Необходимость вызова СМП	NEED_CALL_AMBULANCE																						
2	Необходимость вызова неотложной помощи	NEED_CALL_EMERGENCY_MEDICAL_CARE																						
3	Пациент явился на прием в МО	PATIENT_CAME_TO_THE_APPOINTMENT																						
4	Пациент ранее обслужен	HOME_VISIT_HAS_BEEN_SERVED																						
5	Дублирование	HOME_VISIT_DUPLICATE																						

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
					вызова врача на дом 6 Ошибочно создан вызов врача на дом 7 Отказ пациента от вызова врача на дом 8 Пациент не найден по указанному адресу 9 Пациент госпитализирован 10 Вызов врача на дом не может быть принят в нерабочее время сервиса 11 Другое HOME_VISIT_WRONG_CREATED PATIENT_DECLINED PATIENT_NOT_FOUND_ON_ADDRESS PATIENT_IS_HOSPITALIZED HOME_VISIT_CANT_BE_REGISTERED ANOTHER	
Описание причины отмены записи	reason_description	reason_description	Н	string	Обязательно для заполнения для записей, у которых reason = ANOTHER – Другое	Да
Источник записи	tag_source	tag_source	Н	string	Возможные значения: EPGU – ЕПГУ; REG – регистратура; INFOMAT - инфомат; DOCTOR - врач; CALL – колцентр; OTHER_WEB – другой источник записи	Да
Уникальный идентифи-	resource_id	resource_id	Н	string	Уникальный идентификатор ресурса (тип ресурса=	Да

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
катор ресурса					SPECIALIST) (должен принадлежать назначенной МО mo_id). Обязательно для заполнения для записей, у которых status_code = 5000 (RECORDED) – Вызов на дом подтвержден; 5020 (SUCCESS) – Вызов выполнен, услуга оказана. Для прочих статусов – в случае наличия.	
Временной интервал визита врача, с ... ч	visit_date_start	visit_date_start	Н	string	Обязательно для заполнения для записей, у которых status_code = 5000 (RECORDED) – Вызов на дом подтвержден; 5020 (SUCCESS) – Вызов выполнен, услуга оказана. Для прочих статусов – в случае наличия.	Да
Временной интервал визита врача, по ... ч	visit_date_end	visit_date_end	Н	string	Обязательно для заполнения для записей, у которых status_code = 5000 (RECORDED) – Вызов на дом подтвержден; 5020 (SUCCESS) – Вызов выполнен, услуга оказана. Для прочих статусов – в случае наличия.	Да
ID адреса вызова ФИАС	address_fias_guid	address_fias_guid	О	string	Глобальный уникальный идентификатор адресного объекта ГАР ФИАС	Нет
Адрес вызова	address_full	address_full	О	string	Адрес вызова в текстовом виде	Да
Идентификатор улицы ФИАС	address_street_fias_guid	address_street_fias_guid	О	string	Глобальный уникальный идентификатор адресного объекта ГАР ФИАС	Нет
Номер дома	address_house	address_house	О	string		Нет
Номер квартиры	address_flat	address_flat	Н	string		Нет
Контактный телефон	patient_phone	patient_phone	О	string	Сведения из формы заявки на вызов врача на ЕПГУ	Да
Дополнительной информации	patient_com	patient_com	Н	string		Нет

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
мации для получения доступа к помещению, в котором находится пациент (номер подъезда, этаж, код домофона)	ment	ment				
Жалобы пациента, основание для вызова врача на дом	patient_symptoms	patient_symptoms	Н	string		Да
Дата создание записи	create_ts	create_ts	Н	string		Нет
Дата изменения записи	update_ts	update_ts	Н	string		Нет
Дата заявки на вызов врача на дом	home_visit_request_ts	home_visit_request_ts	Н	string		Нет
Желаемая дата вызова	desired_date	desired_date	Н	string	Формат: YYYY-MM-DD	
Жалобы	complaints_emis	complaints_emis	Н	array [string]	Код жалоб для ЕМИС Столбец code из справочника «house_call_complaint»	
Симптомы для ЕМИС	symp_emis	symp_emis	Н	array [string]	Код симптомов для ЕМИС Столбец symptom_code из справочника «house_call_complaint» Возможные значения: PAIN_SYNDROME - Болевой синдром TEMPERATURE - Температура (гипертермический синдром) GIDDINNES - Головокружения INJURIES - Травмы HEADACHE - Головная боль DETERIORATION - Ухудшение состояния пациента после оказания МП	

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
					VOMITING - Рвота RASH - Сыпь SWELLING - Отеки ALLERGY - Аллергическая реакция OTHER – Иное	
Этаж	floor	floor	Н	string	Этаж	
Подъезд	entrance	entrance	Н	string	Подъезд	
Домофон	intercom	intercom	Н	string	Домофон	
Тип вызова	type_call	type_call	Н		Возможные значения: HOME_VISIT – Вызов на дом ACTIVE - Актив PATRONAGE - Патронаж EMERGENCY_CARE - Неотложная помощь	
Приоритет	priority	priority	Н	boolean		

Таблица 14 – Locations - Участки

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Уникальный идентификатор записи (участка)	id	id	O	string	GUID Формат: нижний регистр, через дефис	
Номер участка	number	number	O	string(45)	GUID Формат: нижний регистр, через дефис	
Тип участка	type	type	O	string(45)	THERAPY - терапевтический (в том числе цеховой) GENERALDOC - врача общей практики(семейного врача) COMPLEX – комплексный GYNECOLOG – акушерский ASCRIBED- Приписной PEDIATRICS – Педиатрический DENTIST – Стоматологический	
Идентификатор мед организации, к которой относится участок	mo_id	mo_id	O	string	GUID Формат: нижний регистр, через дефис	
OID структурного подразделения	depart_oid	depart_oid	O	string	Значение уник. идентификатора МО или СП МО. Должно соответствовать значению в ФРМО	
Дата создание записи	create_ts	create_ts	O	string	Формат:YYYY-MM-DDTHH:MI:SS FORMAT RFC3339 (ISO8601)	
Дата изменения записи	update_ts	update_ts	H	string	Формат:YYYY-MM-DDTHH:MI:SS FORMAT RFC3339 (ISO8601)	

Таблица 15 – Resource Location – Ресурсы закрепленные за участками

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Уникальный идентификатор записи (прикрепления)	id	id	O	string	GUID Формат: нижний регистр, через дефис	
Идентификатор ресурса	resource_id	resource_id	O	string(45)	GUID Формат: нижний регистр, через дефис	
Идентификатор участка	location_id	location_id	O	string(45)	Уникальный идентификатор участка. id выгружаемого участка должен совпадать с id участка в МИС GUID Формат: нижний регистр, через дефис	
Дата начала прикрепления ресурса	beg_date	beg_date	O	string	Формат: YYYY-MM-DDTHH:MI:SS ФОРМАТ RFC3339 (ISO8601)	
Дата окончания прикрепления ресурса	end_date	end_date	O	string	Формат: YYYY-MM-DDTHH:MI:SS ФОРМАТ RFC3339 (ISO8601)	
Дата создание записи	create_ts	create_ts	O	string	Формат: YYYY-MM-DDTHH:MI:SS ФОРМАТ RFC3339 (ISO8601)	
Дата изменения записи	update_ts	create_ts	H	string	Формат: YYYY-MM-DDTHH:MI:SS ФОРМАТ RFC3339 (ISO8601)	
Статус врача на участке (основной / замещающий)	is_primary	is_primary	H	boolean	Значение TRUE, если: Врач является основным на участке Значение FALSE, если: Врач является замещающим на участке	

Таблица 16 – Observation – Записи по карте на диспансерное наблюдение

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Уникальный идентификатор записи	id	id	O	string	GUID Формат: нижний регистр, через дефис	Да
Идентификатор пациента	patient_id	patient_id	O	string(45)	GUID Формат: нижний регистр, через дефис	Да
Идентификатор ресурса (врача)	resource_id	resource_id	O	string(45)	GUID Формат: нижний регистр, через дефис	Да
Информация о статусе карты	status	status	H	boolean	Значение TRUE, если: Карта учета актуальная. То есть, если дата начала диспансерного наблюдения пациента меньше или равно текущей дате и дата окончания диспансерного наблюдения пациента не заполнено или больше текущей даты. Карта учета не удалена. Если у пациента существует несколько карт учета, то значение TRUE проставляется всем актуальным картам.	Да
Возрастная категория пациента	type_age	type_age	H	string(45)	Возможные значения: ADULT – Если 1 (Взрослое население) CHILD – Если 2 (Детское население) NULL – Если 3 (Смешанный прием) или Null	Да
Повод для обращения (диагноз)	d_observation_reason	d_observation_reason	H	string(45)	Код и наименование диагноза	Нет
Дата создания записи	create_ts	create_ts	O	string	Формат: YYYY-MM-DDTHH:MI:SS FORMAT RFC3339 (ISO8601)	Нет
Дата изменения записи	update_ts	update_ts	H	string	Формат: YYYY-MM-DDTHH:MI:SS FORMAT RFC3339 (ISO8601)	Нет
Врач, снявший с учета	position_end_id	position_end_id	H	string	GUID	Нет

Наименование	Атрибут	Имя атрибута в топике Kafka	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
					Формат: нижний регистр, через дефис	
Дата постановки на учет	start	start	О	string	Формат: YYYY-MM-DDTHH:MI:SS ФОРМАТ RFC3339 (ISO8601)	Да
Дата снятия с учета	end	end	Н	string	Формат: YYYY-MM-DDTHH:MI:SS ФОРМАТ RFC3339 (ISO8601)	Нет

Приложение 2. Структура ответа сервиса взаимодействия с ПО «Витрина данных»

Элемент		Описание
result		Результат ответа
isSuccess		Статус результата
error		Описание ошибки: - в случае успешного выполнения операции (StatusCode: 200), поле «error» заполняется null; - в иных случаях: «errorCode», «errorMessage»
	errorMessage	Описание ошибки
	errorCode	Код ошибки
errors		Описание ошибки: - в случае успешного выполнения операции (StatusCode: 200), поле «errors» заполняется null; - в иных случаях: «id», «ErrorCode», «ErrorMessage»
	id	Идентификатор записи с ошибкой
	errorMessage	Описание ошибки
	errorCode	Код ошибки
sendedGuids		Идентификаторы записей без ошибок
statusCode		Код ответа: 200 - Выполнено успешно 400 - Формат переданных сведений некорректен и не соответствует требованиям 401 - Требуется авторизация 500 - Ошибка обработки запроса 503 – Ведутся технические работы

Приложение 3. Взаимодействие с брокером сообщений Kafka

1. Обзор Kafka топиков

1.1 Топики для подписания сервисов ЕМИС

Ниже приведен список доступных топиков:

Таблица 1 – Список топиков для подписания сервисов ЕМИС

Наименование топика в Kafka	Содержимое топика
emis_attachment	Прикрепления
emis_book	Бирки
emis_home_visit	Вызовы врачей на дом
emis_locations	Участки
emis_mo	Медицинские организации
emis_observation	Карта диспансерного наблюдения
emis_patient	Пациенты
emis_profilecode_resource	Профиль ресурса
emis_referral	Направление
emis_resource	Ресурсы МО
emis_resource_location	Ресурсы участка
emis_service	Услуги
emis_slot	Слоты
emis_unaccessible_period	Периоды недоступности
emis_waiting_list	Листы ожидания
emis_waiting_list_entity	Записи в листе ожидания

1.2 Топики для подписания сервисов МИС

Ниже приведен список доступных топиков:

Таблица 2 – Список топиков для подписания сервисов МИС

Наименование топика в Kafka	Содержимое топика
non_emis_attachment	Прикрепления
non_emis_book	Бирки
non_emis_home_visit	Вызовы врачей на дом
non_emis_locations	Участки
non_emis_mo	Медицинские организации
non_emis_observation	Карта диспансерного наблюдения
non_emis_patient	Пациенты

non_emis_profilecode_resource	Профиль ресурса
non_emis_referral	Направление
non_emis_resource	Ресурсы МО
non_emis_resource_location	Ресурсы участка
non_emis_service	Услуги
non_emis_slot	Слоты
non_emis_unaccessible_period	Периоды недоступности
non_emis_waiting_list	Листы ожидания
non_emis_waiting_list_entity	Записи в листе ожидания

Подробное описание полей сущностей, передаваемых в топиках представлено в **Приложении 1** настоящего регламента.

2. Требования к чтению сообщений из Kafka

Для обеспечения корректности и атомарности операций при работе с Kafka, необходимо соблюдать следующее требование: чтение только закоммиченных сообщений.

Consumer должен быть настроен на чтение только закоммиченных сообщений. Это достигается путем установки параметра `IsolationLevel = IsolationLevel.ReadCommitted`.

Данная настройка гарантирует, что Consumer не будет читать технические или незакоммиченные сообщения, что исключает возможность обработки некорректных или промежуточных данных.

3. Примеры пользовательских сценариев

3.1. МИС получает входящие сообщения из Kafka по топикам non_emis_*.

3.1.1. МИС читает заголовок (headers), определяет принадлежность сообщения к МО по mo_oid.

3.1.2. МИС читает Key/Value:

Key — идентификатор сущности

Value — JSON (создание/обновление) или null (удаление)

sys_op в header: 0 — добавление/обновление, 1 — удаление

3.2. non_emis_mo (подразделения МО)

3.2.1 МИС определяет наличие идентификатора подразделения в своей системе.

3.2.2 Если данные найдены, сообщение считается обработанным.

3.2.3 Если данные отсутствуют, МИС сопоставляет подразделение по branch_id и сохраняет внешний mo_id на стороне сервиса интеграции.

3.3. non_emis_resource (ресурсы МО)

3.3.1 МИС определяет наличие идентификатора ресурса в своей системе.

3.3.2 Если данные найдены, сообщение считается обработанным.

3.3.3 Если данные отсутствуют, МИС сопоставляет ресурс по snils, post_id, mo_id и сохраняет внешний resource_id.

3.4. non_emis_slot (слоты)

3.4.1 МИС определяет наличие идентификатора слота. Если данные найдены, МИС сверяет ключевые параметры (например: status, tag_type, tag_age) и при расхождении обновляет слот.

Если данные отсутствуют, см. п. 3.4.2–3.4.4.

3.4.2. МИС определяет resource_id слота и сопоставляет с внутренним идентификатором специалиста.

3.4.3. МИС создает слот по входящим параметрам (время приема visittime, тип tag_type, возраст tag_age).

3.4.4. МИС помечает слот как обработанный из Kafka во избежание дублирования.

3.5. non_emis_book (бирки)

3.5.1. МИС определяет наличие идентификатора бирки. Если данные найдены, МИС сверяет параметры (например: status_code) и при расхождении изменяет запись пациента по слоту. Если данные отсутствуют, см. п. 3.5.2–3.5.4.

3.5.2. МИС определяет slot_id, сопоставляет с внутренним слотом.

3.5.3. МИС обновляет внутренний слот и записывает пациента (по patient_id), выставляет источник записи (tag_source).

3.5.4. МИС сохраняет идентификатор бирки и помечает ее как успешно обработанную.

3.6. non_emis_unaccessible_period (периоды недоступности)

3.6.1. МИС определяет наличие идентификатора периода недоступности.

Если данные найдены, МИС сверяет ключевые параметры (например: date_from, date_to, status) и при расхождении обновляет период.

Если данные отсутствуют, см. п. 3.6.2–3.6.3.

3.6.2. МИС сопоставляет resource_id (или связанный объект периода) с внутренним идентификатором специалиста/ресурса.

3.6.3. МИС создает период недоступности по входящим данным и помечает его как обработанный из Kafka.

3.7. non_emis_home_visit (вызовы врача на дом)

3.7.1. МИС определяет наличие идентификатора вызова в своей системе.

Если данные найдены, сообщение считается обработанным.

Если данные отсутствуют, МИС сопоставляет вызов по ключевым параметрам (например: patient_id, resource_id, date_from/date_to) и сохраняет внешний home_visit_id.

3.7.2. МИС сверяет ключевые параметры (например: status, date_from, date_to, resource_id) и при расхождении обновляет вызов.

3.7.3. МИС помечает вызов как обработанный из Kafka.

Если Value = null и sys_op = 1: МИС удаляет или помечает соответствующую сущность (mo/resource/slot/book/unaccessible_period/home_visit) как неактуальную.

4. Формат сообщений

Сообщения в Kafka состоят из трех основных компонентов: ключа (Key), значения (Value) и заголовка (Header).

4.1. Ключ (Key)

Ключ является обязательным элементом сообщения и должен присутствовать в каждом сообщении.

Формат: строка (string).

Содержание: идентификатор сущности.

Пример ключа:

```
9be23f2b-d56b-48b9-5c5d-f14aca7ce015
```

4.2. Значение (Value)

Формат: строка (string), которая может быть null или содержать JSON-объект.

Если значение равно null, это указывает на удаление сущности.

Если значение не равно null, это указывает на добавление или обновление сущности. Внутри содержится JSON-объект с данными.

Пример JSON-объекта:

```
{
  "id": "9be23f2b-d56b-48b9-5c5d-f14aca7ce015",
  "patient_id": "89579482-e678-4718-aac1-16c66e06a7c8",
  "slot_id": "9d1cc40a-dde7-4c0c-2aed-7aee47d46d49",
  "status_code": 5040,
  "tag_source": "OTHER_WEB",
  "type": "APPOINTMENT",
  "referral_id": "string",
  "create_ts": "2023-11-20T08:00:00",
  "update_ts": "2025-02-21T09:54:38"
}
```

4.3. Заголовок (Header)

Формат: JSON-объект, содержащий техническую информацию.

Поля:

mo_oid: Идентификатор медицинской организации (oid)

session: Идентификатор сессии (string).

Type: Тип модели (string).

Sys_op: Тип операции (string). Значения: 0 — добавление, 1 — удаление

Пример JSON-объекта заголовка:

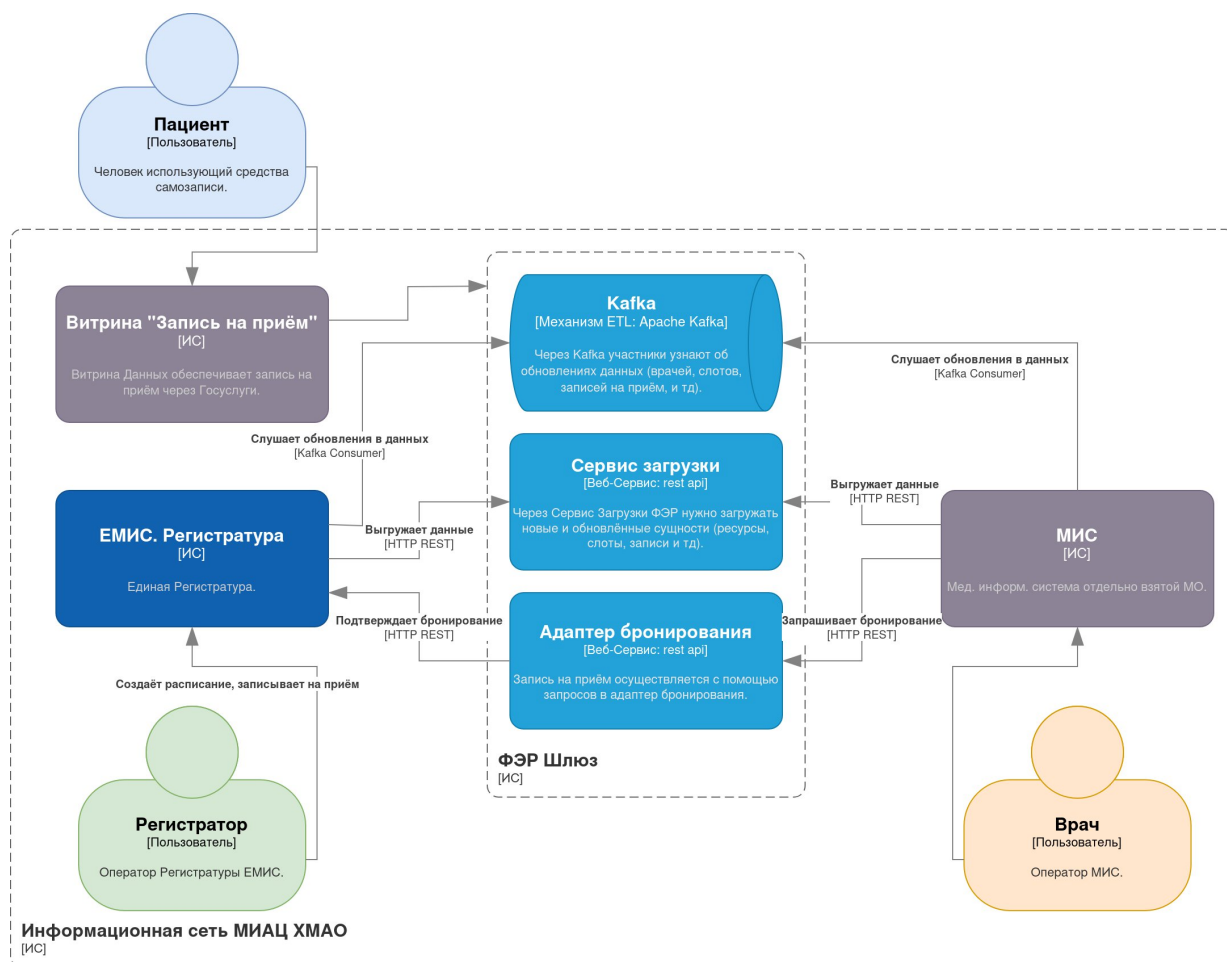
```
{
  "mo_oid": "1.2.643.5.1.13.13.12.2.86.9001",
  "session": "07a2996e-e1b5-4649-a027-86b560f3f4a6",
  "type": "BookEmisKafkaSendingModel",
  "sys_op": "1"
}
```

5. Интеграция сторонних МО с Регистратурой ЕМИС через топики Kafka

5.1 Общая схема взаимодействия.

Информационное взаимодействие выстроено через информационную систему ФЭР Шлюз:

- 1) МИС МО подписывается на топики Kafka для получения обновлений данных
- 2) н-р, подписались на топик non_emis_slot – начали получать новые, обновлённые и удалённые слоты
- 3) При записи на приём МИС МО обращается в Адаптер бронирования для подтверждения записи
- 4) забронировали слот – обратились в адаптер бронирования, ЕМИС подтвердил, из Kafka пришли обновлённые слот и факт записи



5.2 Действия со стороны МИС

5.2.1 Процесс бронирования

При бронировании слота внутри МИС, МИС нужно обратиться к следующим энд-поинтам сервиса бронирования:

- /MIS/booking/book – для подтверждения бронирования
- /MIS/booking/cancel – для отмены бронирования

См. Сваггер для уточнения параметров.

5.2.2 Вызовы врача на дом

МИС изменяет статусы вызова врача на дом. Нужно сообщить об изменении статуса в ЕМИС: выгрузить вызов с обновлённым статусом через Сервис Загрузки

6. Адреса Контуров

Таблица 3 – Адреса контуров

Название ИС	Адрес	Адрес тестового контура
Адаптер бронирования	10.86.6.189:2101 (<u>Swagger</u>)	10.86.6.237:2101 (<u>Swagger</u>)
Сервис загрузки	10.86.6.189:2100 (<u>Swagger</u>)	10.86.6.237:2100 (<u>Swagger</u>)
Кластер Kafka Шлюза ФЭР 2.0 (брокеры)	10.86.6.64:9092(broker_1: 29092) 10.86.6.65:9092(broker_3: 29092) 10.86.6.66:9092(broker_2: 29092)	10.86.6.237:9092 - Broker_1 10.86.6.237:9093 - Broker_2 10.86.6.237:9094 - Broker_3

Приложение 4. Настройка передачи полей **origin_source** и **last_change_source** для организации учета бронирования

1. Инициализация источника в адаптере бронирования

Адаптер бронирования определяет детальный источник по endpoint/клиенту и передает его в поле **booking_source**.

В **booking_source** из адаптера могут приходить значения:

- EMIS
- MIS
- EPGU
- MAX_BOT
- VOICE_ASSISTANT
- ТМК

МИС/ЕМИС должны принять это значение, корректно распарсить и при выгрузке сущности **book** в ФЭР Шлюз передать:

- **last_change_source** = детальный источник из **booking_source**;
- **tag_source** = только регламентное значение.

Обязательное правило: Источник текущего события всегда передается в ФЭР Шлюз через **last_change_source**. Поле **origin_source** в запросах **book** не используется как источник события.

Таблица 1 - Таблица сопоставления

booking_source из адаптера	tag_source при выгрузке `book`	last_change_source при выгрузке `book`
EPGU	EPGU	EPGU
MAX_BOT	OTHER_WEB	MAX_BOT
VOICE_ASSISTANT	OTHER_WEB	VOICE_ASSISTANT
ТМК	OTHER_WEB	ТМК
MIS	REG или DOCTOR	MIS
EMIS	REG	EMIS

Примечание по **tag_source**:

- использовать только регламентные значения (REG, DOCTOR, EPGU, OTHER_WEB);
- не передавать детальные источники (MIS, EMIS, MAX_BOT, VOICE_ASSISTANT, ТМК) в **tag_source**.

2. Как действовать МИС/ЕМИС как подтверждающей системе

Если бронь/изменение/отмена выполняется через МИС или ЕМИС и они являются подтверждающей системой:

1. Определить фактический источник текущего действия по полю booking_source из пришедшего из адаптера бронирования json:

- операция выполнена в МИС -> last_change_source = MIS;
- операция выполнена в ЕМИС/КЦ -> last_change_source = EMIS.

2. Заполнить tag_source по регламенту канала внутри системы:

- действие регистратуры -> REG;
- действие врача -> DOCTOR.

3. Передать в ФЭР Шлюз book с корректной парой полей:

- tag_source (регламентное значение);
- last_change_source (детальный источник текущего события).

3. Единые требования для всех операций

Для каждой операции с book (создание, изменение, отмена, повторная выгрузка/upsert):

- last_change_source обязателен;
- значение last_change_source должно соответствовать источнику текущего события;
- tag_source должен оставаться в рамках регламента.

4. Минимальные примеры

Создание через МИС (действие регистратора)

```
{
  "tag_source": "REG",
  "last_change_source": "MIS"
}
```

Изменение через бот

```
{
  "tag_source": "OTHER_WEB",
  "last_change_source": "MAX_BOT"
}
```

Отмена через голосового помощника

```
{
  "tag_source": "OTHER_WEB",
  "last_change_source": "VOICE_ASSISTANT"
}
```

Создание через ЕПГУ

```
{
  "tag_source": "EPGU",
}
```

```
"last_change_source": "EPGU"
```

```
}
```