

ООО "РТ МИС"

ЕДИНАЯ ЦИФРОВАЯ ПЛАТФОРМА.МИС 3.0

(ЕЦП.МИС 3.0)

Руководство администратора

Подсистема "Взаимодействие с внешними голосовыми ботами (в т.ч. технологии ИИ) 122" 3.0.3

Содержание

1	Введение.....	5
1.1	Область применения.....	5
1.2	Краткое описание возможностей	5
1.3	Уровень подготовки администратора.....	5
1.4	Перечень эксплуатационной документации, с которым необходимо ознакомиться администратору	5
2	Назначение и условия применения	6
2.1	Условия, при соблюдении которых обеспечивается применение средства автоматизации	6
3	Подготовка к работе.....	7
3.1	Состав и содержание дистрибутивного носителя данных.....	7
3.2	Порядок запуска Системы	7
3.3	Смена пароля.....	11
3.4	Контроль срока действия пароля	12
3.5	Порядок проверки работоспособности.....	12
4	Описание операций	14
5	Общая информация	15
5.1	Общее описание.....	15
5.1.1	Кэширование запросов.....	15
5.1.2	HTTP ошибки.....	15
5.1.3	Общие параметры для всех методов.....	15
5.1.4	Описание параметров метода	16
5.2	Схема взаимодействия	16
5.3	Описание общих кодов ошибок	17
6	Авторизация.....	18
6.1	Авторизация в PHP API. Авторизация в системе	18
6.2	Авторизация в SWAN API. Метод авторизации пользователя СМП.....	18
7	Информирование пациентов о результатах анализов.....	20
7.1	Передача списка пациентов, получивших услуги лабораторных исследований	20
8	Запись по направлению	22
8.1	Получение списка направлений	22
9	Запись на вакцинацию	23

9.1	Получение расписания на вакцинацию в МО прикрепления пациента и ближайшие МО	23
9.2	Запись пациента на приём службы	24
10	Описание методов работы с человеком	26
10.1	Получение списка физических лиц, по ключевым параметрам.....	26
10.2	Получение информации по человеку	26
10.3	Создание человека	28
10.4	Редактирование данных человека	33
10.5	Получение записей на прием к врачу	36
10.6	Получение прикреплений пациента.....	37
10.7	Получение основного прикрепления пациента	38
11	Запись на прием к врачу	41
11.1	Получение должности по наименованию	41
11.2	Метод получения прикреплений пациента	41
11.3	Получение МО по идентификатору.....	43
11.4	Получение основного прикрепления пациента	43
11.5	Получение информации о докторе	45
11.6	Получение списка специальностей в МО.....	49
11.7	Получение списка мест работы по специальности и МО, или по должности и МО.....	49
11.8	Получение свободных дат приема	51
11.9	Получение свободного времени приема	52
11.10	Запись пациента на прием.....	53
12	Отмена записи на прием.....	55
12.1	Отмена записи на прием, услугу по идентификатору бирки	55
12.2	Изменение статуса записи на прием	55
13	Запись на диспансерное наблюдение	56
13.1	Получение свободных дат приема службы диспансерного наблюдения.....	56
13.2	Получение свободного времени приёма службы диспансерного наблюдения	57
13.3	Получение списка услуг диспансеризации, на которые можно оформить запись.....	58
14	Обзвон пациентов записанных на прием.....	59
14.1	Получение списка прошедших и предстоящих занятых бирок в МО за заданный временной промежуток.....	59
14.2	Получение списка предстоящих приемов пациента в МО	60

14.3	Получение списка пациентов, подходящих для вакцинации от Covid–19	61
15	Вызов врача на дом	63
15.1	Добавление вызова врача на дом	63
15.2	Получение адреса	69
15.3	Получение списка параметров адреса	72
16	Работа с направлениями	74
16.1	Создание направления.....	74
17	Лист ожидания.....	82
17.1	Добавление записи в лист ожидания	82
17.2	Изменение статуса записи в листе ожидания	83
17.3	Изменение листа ожидания	83
17.4	Получение листа ожидания	84
18	Общие методы	86
18.1	Поиск человека по атрибутам.....	86
19	Вызов СМП SWAN API.....	91
19.1	Метод добавления талона вызова	91
20	Аварийные ситуации	95
20.1	Описание аварийных ситуаций	95
20.2	Действия в случае несоблюдения условий выполнения технологического процесса	96

1 Введение

1.1 Область применения

Настоящий документ описывает порядок работы с подсистемой "Взаимодействие с внешними голосовыми ботами (в т.ч. технологии ИИ) 122" 3.0.3 (далее – Подсистема, подсистема), являющейся частью Единой информационной системы здравоохранения (далее – Система, система).

1.2 Краткое описание возможностей

Подсистема предназначена для настройки параметров работы Системы на уровнях региона и медицинских организаций.

1.3 Уровень подготовки администратора

К администраторам Подсистемы предъявляются следующие требования:

- глубокое понимание Подсистемы на уровне технологий работы;
- знание основ администрирования;
- знание основ администрирования реляционных баз данных, поддерживающих клиент-серверный режим;
- навыки реализации различных режимов работы операционных систем;
- навыки администрирования учетных записей пользователей Системы.

1.4 Перечень эксплуатационной документации, с которым необходимо ознакомиться администратору

Перед началом работы администраторам рекомендуется ознакомиться с положениями данного Руководства администратора в части своих функциональных обязанностей.

2 Назначение и условия применения

2.1 Условия, при соблюдении которых обеспечивается применение средства автоматизации

Доступ к функциональным возможностям и данным Подсистемы реализуется посредством веб-интерфейса. Работа пользователей Подсистемы осуществляется на единой базе данных ЦОД. Подсистема доступна из любой организации (участника информационного обмена) при наличии канала связи в круглосуточном режиме.

Работа в Подсистеме выполняется через автоматизированные рабочие места персонала (в соответствии с местом работы, уровнем прав доступа к функциональным возможностям и данным Системы).

Настройка рабочего места (создание, настройка параметров работы в рамках МО, предоставление учетной записи пользователя) выполняется пользователем АРМ администратора МО. Настройка общесистемных параметров работы, конфигурация справочников выполняется пользователем АРМ администратора ЦОД.

Настройка внутрисистемных уведомлений пользователям Системы на уровне МО выполняется пользователем АРМ администратора МО. Расчет статистических показателей на основании структурированных исходных данных выполняется пользователем АРМ администратора МО.

3 Подготовка к работе

3.1 Состав и содержание дистрибутивного носителя данных

Подсистема передается в виде функционирующего комплекса на базе средств вычислительной техники.

Подсистема разворачивается Исполнителем.

Работа в Подсистеме возможна через следующие браузеры (интернет-обозреватели):

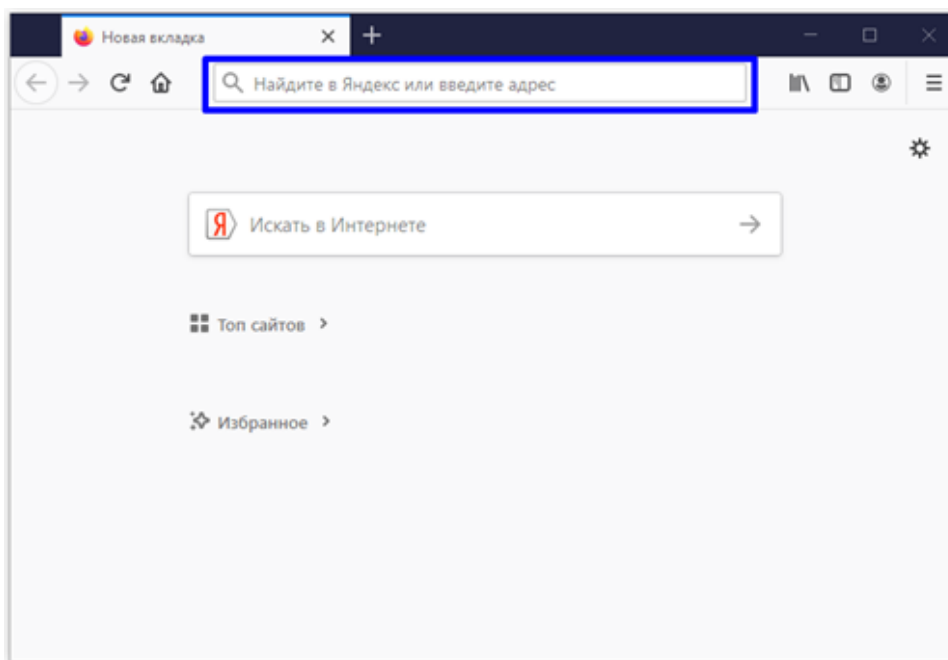
- Mozilla Firefox (рекомендуется);
- Google Chrome.

Перед началом работы следует убедиться, что установлена последняя версия браузера. При необходимости следует обновить браузер.

3.2 Порядок запуска Системы

Для входа в Систему необходимо выполнить следующие действия:

- а) Запустите браузер, например, "Пуск" → "Программы" → **Mozilla Firefox**.
Отобразится окно браузера и домашняя страница.



- б) Введите в адресной строке обозревателя IP-адрес страницы портала, нажмите клавишу **Enter**. На главной странице Системы отобразится перечень программных продуктов.

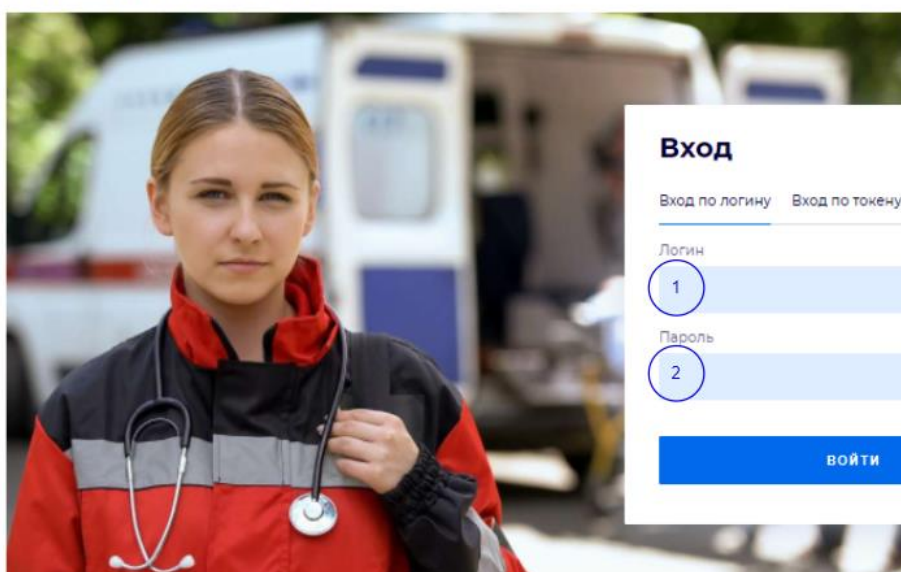
Примечание – Адрес для подключения предоставляется администратором. Если страница Системы установлена в качестве домашней страницы, то она отобразится сразу после запуска браузера.

Для удобства использования рекомендуется добавить адрес портала в закладки интернет-обозревателя, и/или сделать страницу портала Системы стартовой страницей.



Уважаемые пользователи!
Телефон службы технической поддержки ЕИСЗ ПК: 8 800 350 56 52
[Подробнее](#)

Единая информационная система здравоохранения Пермского края



Авторизация в Системе возможна одним из способов:

- с использованием логина и пароля;
- с помощью ЭП (выбора типа токена и ввод пароля);
- с помощью учетной записи ЕСИА.

1. Способ:

- Введите логин учетной записи в поле "**Имя пользователя**" (1).
- Введите пароль учетной записи в поле "**Пароль**" (2).
- Нажмите кнопку "**Войти в систему**".

2. Способ:

- Перейдите на вкладку "Вход по токenu":

Вход

Вход по логину

Вход по токenu

Вход через ЕСИА

Тип токена

AuthApi - eToken ГОСТ



ПИН-код

ВХОД ПО КАРТЕ

- Выберите тип токена.
- Введите пароль от ЭП в поле "**ПИН-код/Сертификат**" (расположенное ниже поля "Тип токена"). Наименование поля зависит от выбранного типа токена.
- Нажмите кнопку "**Вход по карте**".

Примечания

- 1 На компьютере Пользователя предварительно должно быть установлено и запущено программное обеспечение для выбранного типа токена.
- 2 Предварительно может потребоваться установить сертификаты пользователей администратором системы в программном обеспечении выбранного типа токена.

При неправильном вводе имени пользователя и (или) пароля отобразится соответствующее сообщение. В этом случае необходимо повторить ввод имени пользователя и (или) пароля.

3. Способ:

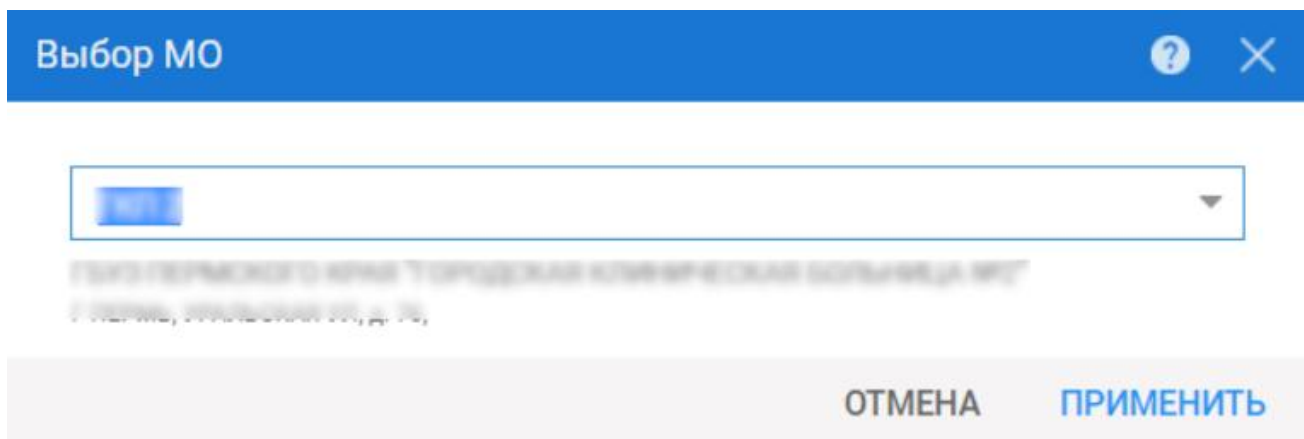
- Перейдите на вкладку "Вход через ЕСИА". Будет выполнен переход на страницу авторизации через ЕСИА.
- Введите данные для входа, нажмите кнопку "**Войти**".

Примечания

- 1 Для авторизации с помощью токена на компьютере Пользователя предварительно должно быть установлено и запущено программное обеспечение для выбранного типа токена. Может потребоваться установка сертификатов пользователей администратором системы в программном обеспечении выбранного типа токена.
- 2 Для авторизации через ЕСИА учетная запись пользователя должна быть связана с учетной записью человека в ЕСИА. Учетная запись пользователя должна быть включена в группу "Авторизация через ЕСИА".

При неправильном вводе имени пользователя и (или) пароля отобразится соответствующее сообщение. В этом случае необходимо повторить ввод имени пользователя и (или) пароля.

- Отобразится форма выбора МО.



Укажите необходимую МО и нажмите кнопку "**Применить**".

- Отобразится форма выбора АРМ по умолчанию.

АРМ/МО	Подразделение / Отделение / Служба	Должность	Расписание
АРМ администратора ЛЛО	Администратор ЛЛО		
АРМ администратора МО (Е...			
АРМ администратора ЦОД			
АРМ администратора ОУ	ОУ		

ОТМЕНА ПРИМЕНИТЬ

Примечание – Форма отображается, если ранее не было выбрано место работы по умолчанию, или при входе была изменена МО. После выбора места работы, указанный АРМ будет загружаться автоматически после авторизации.

Выберите место работы в списке, нажмите кнопку **"Применить"**. Отобразится форма указанного АРМ пользователя.

3.3 Смена пароля

При регистрации учетной записи администратор присваивает ей временный пароль. При первом входе в систему пользователь должен сменить временный пароль, выданный администратором.

После ввода имени пользователя, пароля и нажатия кнопки **"Войти в систему"** выполняется проверка актуальности пароля, как временного, так и постоянного.

Если истек срок действия временного пароля, то отобразится сообщение пользователю: "Истек срок действия временного пароля. Обратитесь к Администратору системы". Далее процесс аутентификации не производится.

Если временный пароль прошел проверку на актуальность, на форме отображаются поля для смены пароля. Рядом с полями отобразится подсказка с требованиями к паролю (указывается минимальная длина и допустимые символы).

При смене временного пароля на постоянный (при первом входе в систему) выполняется проверка на соответствие пароля установленным требованиям безопасности (минимальная длина, пользовательский пароль должен отличаться от временного на указанное количество символов и т.д.).

Вход в систему возможен, если введен актуальный временный пароль, новый пароль соответствует всем требованиям (требования к паролю указаны в параметрах системы), значения полей "Новый пароль" и "Новый пароль еще раз" идентичны. В процессе ввода нового пароля рядом с полем должна отобразиться зеленая галочка, если введенный пароль удовлетворяет всем требованиям. В процессе подтверждения нового пароля рядом с полем "Новый пароль еще раз" отобразится зеленая галочка, если значения полей "Новый пароль" и "Новый пароль еще раз" идентичны.

При входе в систему происходит сохранение нового пароля.

3.4 Контроль срока действия пароля

При каждом входе в систему выполняется проверка срока действия пароля. Срок действия пароля определяется настройками системы (рассчитывается от даты создания пароля).

За несколько дней до истечения срока действия пароля при входе в систему выводится информационное сообщение "До истечения срока действия пароля осталось %кол-во дней% дней. Пароль можно сменить в личном кабинете".

При входе в систему в последний день актуальности пароля, на форме авторизации отобразятся поля для смены пароля.

Вход в систему возможен, если введен верный старый пароль, а новый пароль соответствует всем требованиям (требования к паролю указаны в параметрах системы), значения поле "Новый пароль" и "Новый пароль еще раз" идентичны.

В процессе ввода нового пароля рядом с полем отобразится зеленая галочка, если введенный пароль удовлетворяет всем требованиям.

В процессе подтверждения нового пароля рядом с полем "Новый пароль еще раз" отобразится зеленая галочка, если значения полей "Новый пароль" и "Новый пароль еще раз" идентичны.

При входе в систему происходит сохранение нового пароля.

3.5 Порядок проверки работоспособности

Для проверки работоспособности системы необходимо выполнить следующие действия:

- Выполнить авторизацию в Системе и открыть АРМ.
- Вызвать любую форму.

При корректном вводе учетных данных должна отобразиться форма выбора МО или АРМ, либо АРМ пользователя. При выполнении действий должно не должно отображаться

ошибок, система должна реагировать на запросы пользователя, например, отображать ту или иную форму.

4 Описание операций

Администрирование Системы осуществляется на двух уровнях: уровне МО и уровне Системы. Для администрирования на уровне МО предназначен функционал "Автоматизированное рабочее место администратора МО" (далее "АРМ администратора МО").

Для администрирования на уровне Системы предназначен функционал "Автоматизированное рабочее место администратора центра обработки данных" (далее АРМ администратора ЦОД).

5 Общая информация

5.1 Общее описание

API построен по архитектуре REST с возвратом результатов в формате JSON.

5.1.1 Кэширование запросов

При выполнении GET запросов сервер может в HTTP заголовке Last-Modified передать дату последнего изменения получаемых данных. Клиент может передать эту дату в последующих аналогичных запросах в HTTP заголовке If-Modified-Since, если данные не были изменены, то сервер вернет пустую страницу с HTTP кодом 304 Not Modified.

5.1.2 HTTP ошибки

Общие HTTP ошибки возвращаются HTTP кодами.

401 – не передан или неверный apiKey

404 – обращение к несуществующему методу

5.1.3 Общие параметры для всех методов

Входящие параметры передаются или через строку запроса (query string), или непосредственно записываются в путь(path) URL, если это указано в названии метода:

- apiKey – ключ API;
- sess_id – идентификатор сессии. Не обязателен для методов, не требующих авторизации пользователя;
- offset – смещение для порции записей.

Выходные данные, в случае если возвращен код HTTP 200 представляют JSON объект со следующими полями:

- error_code – код ошибки, см. справочник кодов ошибок;
- error_msg – сообщение с подробностями ошибки;
- count – количество записей в ответе, в случае если ответ неполон;
- offset – смещение текущей возвращаемой порции записей, если оно ненулевое;
- data – массив объектов, зависящих от запроса, может содержать 1 и больше элементов.

5.1.4 Описание параметров метода

- title, содержащий текстовое название параметра;
- type, указывающий тип значения параметра; указывается символ формата, а вслед за ним в квадратных скобках максимальная длина атрибута; символы формата соответствуют вышеописанным обозначениям:
 - T – <текст>;
 - N – <число>;
 - D – <дата> в формате ГГГГ–ММ–ДД;
 - DT – <дата и время> в формате ГГГГ–ММ–ДД чч:мм:сс;
 - V – <время> в формате чч:мм:сс;
 - S – <элемент>; составной элемент, (список/массив), описывается отдельно.
 - description, содержащий текстовое описание параметра;
 - required, указывающий на то, что данный параметр является обязательным (допускается указать несколько символов, например OM);
 - O – обязательный реквизит, который должен обязательно присутствовать в элементе;
 - H – необязательный реквизит, который может, как присутствовать, так и отсутствовать в элементе. При отсутствии, не передается;
 - U – условно–обязательный реквизит. При отсутствии, не передается;
 - M – реквизит, определяющий множественность данных, может добавляться к указанным выше символам.
- Используются русские символы
 - default, содержащий значение параметра по умолчанию.

Пример описания параметра

- title (type, required, default) – description
- email (T[60],OM,1@mail.ru) – электронная почта, уникальная

5.2 Схема взаимодействия

- а) Запрос из сторонней МИС приходит в сервис.
- б) Сервис обрабатывает запрос и вызывает соответствующие методы REST API.
- в) Методы REST API производят необходимые манипуляции с данными (поиск, сохранение, удаление).

Принцип авторизации: авторизация через сервис с последующей авторизацией в Системе.

5.3 Описание общих кодов ошибок

- 0 – успешный запрос, отсутствие ошибки;
- 1 – метод требует авторизации, а сессия не передана;
- 2 – неверный идентификатор сессии, просроченная сессия;
- 3 – не передан какой-то из обязательных параметров, в `error_msg` – расшифровка;
- 4 – ошибка во входящих параметрах, в `error_msg` – расшифровка;
- 5 – попытка изменения данных в БД под демо-пользователем.

6 Авторизация

6.1 Авторизация в PHP API. Авторизация в системе

GET /api/user/login

Авторизация пользователя в системе

Входящие параметры	– Login (T [], O) – логин пользователя – Password (T [], O) – пароль пользователя (постоянный пароль)
Ответ	Успешный ответ – Идентификатор сессии Ошибки – не верный логин или пароль пользователя – нет прав доступа
Пример	Запрос http://<ip>/api/user/login

При авторизации пользователя в системе выполняется проверка наличия группы "Пользователь API". Если у пользователя нет группы, то отобразится ошибка "Нет прав доступа".

Авторизация зависит от используемых механизмов:

- при работе с Rest API применяется базовая (Basic) авторизация:
- с помощью метода GET /api/user/login определяется идентификатор сессии (sess_id);
- в последующих вызовах обязательным входящим параметром является идентификатор сессии.
- при использовании SOAP запросов применяется метод WS-security;
- в секцию <wsse:UsernameToken> всегда передается пара логина и пароля.

6.2 Авторизация в SWAN API. Метод авторизации пользователя СМП

GET /login

При работе с Rest API применяется базовая (Basic) авторизация:

- с помощью метода GET /login определяется идентификатор сессии (session);
- в последующих вызовах прикладных методов обязательным входящим параметром является идентификатор сессии.

Время жизни сессии задается в настройках сервиса API, по умолчанию равно 1800 секунд.

Входящие параметры	– user (Т, О) – логин пользователя; – pass (Т, О) – пароль пользователя; – region (Т, Н) – регион
Ответ	– session (Т, О) – идентификатор сессии; – user (Т, О) – идентификатор пользователя; – medPersonal (Т, О) – идентификатор врача
Пример ответа	<pre>{ "errorCode": 0, "errorMessage": "OK", "rowCount": 0, "responseData": { "session": "2V24Eg-V1smbs-S_m3xnoCF5JdGnVqEP7qfleYkN3pdJGa9NN6O-9pEYksmjkf0G", "user": 359128234877, "medPersonal": 10101000000001 }, "success": true }</pre>
Ошибки	Пример: <pre>{ "errorCode": 401, "errorMessage": "Unauthorized", "description": "unknown user swnt_testapi1 or pass is incorrect", "addition": null, "success": false }</pre>

7 Информирование пациентов о результатах анализов

7.1 Передача списка пациентов, получивших услуги лабораторных исследований

GET api/UslugaTestPersonList

Метод предназначен для передачи списка пациентов, получивших услуги лабораторных исследований

Входящие параметры	– Lpu_id (N, H) – идентификатор МО; – LpuBuilding_id(N, H) – идентификатор подразделения; – LpuSection_id (N, H) – идентификатор отделения; – MedService_id (N, H) – идентификатор службы; – UslugaComplex_id (N, H) – идентификатор исследования; – UslugaTest_CheckDTBeg (DT, H) – начало периода, за который были одобрены результаты исследования; – UslugaTest_CheckDTEnd (DT, H) – окончание периода, за который были одобрены результаты исследования; – PersonAgeFrom (N, H) – начало диапазона возрастов пациентов; – PersonAgeTo(N, H) – окончание диапазона возрастов пациентов; – Limit (N, O) – максимальное количество элементов для получения (по умолчанию 100, максимальное значение – 1000); – Page(N, O) – страница, с которой необходимо получить данные
Ответ	Если запрос обработан без ошибок и найдена хотя бы одна запись в БД, то в ответе будет отправлено: – код состояния: 200; – error_code: 0; – data: – Lpu_id (N, H) – МО; – LpuBuilding_id(N, H) – подразделение; – LpuSection_id (N,H) – отделение; – MedService_id (N, H) – служба; – Person_SurName (S, O) – фамилия пациента; – Person_Name (S, O) – имя пациента; – Person_SecName (S, H) – отчество пациента; – Person_Phone (S, H) – номер телефона пациента; – UslugaTest_Result (N, O) – результат выполнения исследования; – UslugaTest_CheckDT (DT, O) – дата одобрения результата исследования. Если запрос обработан без ошибок, но ни одна запись в БД по условиям запроса не найдена, то в ответе будет отправлено: – код состояния: 200; – error_code: 24001; – error_msg: "По запросу данных не найдено". Если в запросе не указано ни одного параметра, то в ответе будет отправлено: – код состояния: 200; – error_code: 24002; – error_msg: "Не указаны параметры запроса". Если в запросе не заполнен хотя бы один обязательный параметр, то в ответе будет отправлено: – код состояния: 200;

	– error_code: 24003; – error_msg: Необходимо заполнить обязательные параметры запроса: наименование параметра
--	---

8 Запись по направлению

8.1 Получение списка направлений

GET api/EvnDirectionList

Входящие параметры	– Evn_pid (N, H) – идентификатор случая–родителя (движение в КВС, посещения в ТАП); – Person_id (N, H) – идентификатор человека; – PayType_id (N, O) – идентификатор типа оплаты; – EvnDirection_begD (D) – дата начала периода направлений; – EvnDirection_endD (D) – дата окончания периода направления; – DirType integer – список типов направлений; – Post integer – список должностей; – EvnStatus (N) – список статусов направлений; – IsDirFailType – признак: список направлений, имеющих тип причины отмены; – ResponseFull – признак: полный ответ; – Lpu_id (N, H) – идентификатор МО; – Diag_id (N, H) – идентификатор диагноза; – Diag_Code (T, H) – код диагноза в МКБ–10. Контроль: хотя бы один из входящих параметров должен быть заполнен.
Ответ	Успешный ответ – список направлений: – EvnDirection_id (N, O) – идентификатор направления; – Evn_id (N, O) – идентификатор случая выписки направления; – Evn_pid (N, H) – идентификатор случая–родителя (движение в КВС, посещения в ТАП); – Person_id (N, O) – человек; – EvnDirection_Num (N, O) – номер направления; – EvnDirection_setDate (D, O) – дата направления; – DirType_id (N, O) – тип направления; – Post_id – идентификатор должности; – Post_name – наименование должности; – EvnStatus_id – идентификатор статуса; – EvnStatus_Name – наименование статуса; – EvnStatusCause_Name –наименование причины отмены; – EvnStatusCause_insDT – дата добавления причины отмены; – DoctisTemplate_Code (N) – идентификатор шаблона Доктис; – DoctisTemplate_Name (T) – наименование шаблона Доктис

9 Запись на вакцинацию

9.1 Получение расписания на вакцинацию в МО прикрепления пациента и ближайшие МО

GET api/VaccinationTimeTableMedService

Метод предназначен для определения по идентификатору пациента или по адресу и по типу прививки:

- а) МО прикрепления,
- б) если нет прикрепления или по месту прикрепления привиться нельзя, то по адресу, переданному в одном из параметров:
 - в KLStreet_id;
 - KLStreet_id и Address_House;
 - Address_Address

определяется участок, обслуживающий данный адрес, а у участка определяется МО и в ней ищутся службы с бирками,

- в) возвращает адрес МО: населённый пункт, улица, дом, кабинет,
- г) возвращает список бирок: идентификатор бирки, дата и время

Примечание – Если нужна прививка от COVID–19, то во входном параметре необходимо указать VaccinationType_Id=30.

Запрос	– Person_id (N,H) – идентификатор человека. Параметр обязателен если не указаны параметры KLStreet_id и Address_House, или не указан параметр Address_Address; – VaccinationType_Id (N,O) – тип прививки. Запрашивается значение справочника "Типы прививок ФЭР"; – KLStreet_id (N, H) – идентификатор улицы проживания. Параметр обязателен если не указан параметр Person_id или Address_Address; – Address_House (T, H) – номер дома; – TimeTable_begDT (D, H) – начало диапазона поиска свободных ячеек расписания; – TimeTable_endDT (D, H) – конец диапазона поиска свободных ячеек расписания; – Address_Address (T, H) – строка адреса в формализованном виде: – Формат строки: индекс, страна, область, район, город/населённый пункт, улица, номер дома, номер корпуса, номер квартиры; – допустимы форматы:,,,город,улица,номер дома/здания,,; Указанные параметры обязательны. Ведущие/начальные запятые обязательны – LpuSectionAge_Code (N, H) – код типа отделения по возрасту (предназначен для определения возрастной группы прививаемого в случае
---------------	--

	поиска бирки только по параметру Address_Address)
Ответ	Успешный ответ: МО прикрепления или ближайшее МО к МО прикрепления с указанием наименования и полного адреса МО со списком бирок (идентификатор бирки, номер кабинета, дата и время бирки). Если не задан параметр VaccinationType_id – ошибка: не указан тип вакцинации. – VaccinationType_Id (N, H) – тип прививки. Значение справочника "Типы прививок ФЭР"; – VaccinationType_Name (T, H) – наименование прививки; – Lpu_id (N, O) – идентификатор МО удовлетворяющего условиям поиска; – Services(S, O): – MedService_id (N, O) – идентификатор службы; – MedService_Name (T, O) – имя службы; – LpuBuilding_Address (T, O) – адрес фактического местоположения структурного подразделения медицинской организации. Указывается адрес структурного подразделения медицинской организации следующем формате: почтовый индекс, регион, населенный пункт, улица, дом. – MedService_FullName (T, O) – полное наименование службы. Если есть кабинет, то передается формате: "Наименование_медицинской_организации", `наименование_подразделения`, кабинет `номер_кабинета`.". Если кабинета нет, то в формате "Наименование_медицинской_организации", `наименование_подразделения`"; – LpuBuildingOffice_Number (T, O) – номер кабинета; – TimeTable(S, O): – TimeTable_id (N, O) – идентификатор бирки; – TimeTable_begTime (DT, O) – дата и время начала приёма в формате ГГГГ-ММ-ДД ЧЧ:ММ:СС

9.2 Запись пациента на приём службы

POST api/TimeTableMedService/TimeTableMedServiceWrite

Регион: Базовый.

Входящие параметры	– Person_id (N, O) – идентификатор пациента; – MedService_id (N, Y) – идентификатор службы. Является обязательным, если параметр UslugaComplexMedService_id пустой; – UslugaComplexMedService_id (N, Y) – идентификатор связи комплексной услуги и службы. Является обязательным, если параметр MedService_id пустой; – TimeTableMedService_id (N, O) – идентификатор свободной бирки в расписании службы; – EvnQueue_id (N, H) – идентификатор постановки в очередь. Выходные параметры: – Person_id (N, O) – идентификатор пациента; – TimeTableMedService_id (N, O) – идентификатор записи на службу/услугу службы; – EvnQueue_id (N, H) – идентификатор постановки в очередь
Ответ	– Person_id (N, O) – идентификатор пациента; – TimeTableMedService_id (N, O) – идентификатор записи на службу/услугу службы; – EvnQueue_id (N, H) – идентификатор постановки в очередь; – TimeTableStatusLockComment (T,H) – Сообщение о блокировке бирки,

	отображается в ответе если бирка присутствует в БД. Сообщение содержит в себе возможное время разблокировки бирки. Правила: – если во входящих параметрах указан идентификатор постановки в очередь: – в направлении передается TimeTableMedService_id; – Evn_Queue_id удаляется. – должен быть указан один из параметров MedService_id или UslugaComplexMedService_id. Если не указан ни один из них, то ошибка: "Не указан один из обязательных параметров MedService_id или UslugaComplexMedService_id. Если указаны оба параметра, то ошибка: "Должен быть указан один из параметров MedService_id или UslugaComplexMedService_id"; – MedService_id или UslugaComplexMedService_id указывается для того, чтобы удостовериться, что бирка именно для этой службы или услуги. Если указанный идентификатор бирки не совпадает с указанным параметром MedService_id или UslugaComplexMedService_id, то ошибка: "Для параметра <указывается MedService_id или UslugaComplexMedService_id > не верно указан идентификатор бирки "; – если запрос обработан успешно, то в ответе должно быть отправлено: – код состояния: 200; – параметры: – error_code: 0; – data: – Person_id; – TimeTableMedService_id; – EvnQueue_id. Проверка на наличие переданной во входных параметрах бирки в таблице БД по TimeTableMedService_id: – если TimeTableMedService_id есть в БД, записывать на бирку нельзя. Производится подсчет минут от даты и времени, указанных в БД исходя из логики, что бирка может быть заблокирована оператором 15 минут. Соответственно, если TimeTableLock_lockTime=2017-12-23 21:59:34.260000, то бирка станет доступной для записи или занятой в 2017-12-23 22:14:34.260000; – в связи с данной проверкой для бирок, которые нашлись в таблице БД выводится в ответе параметр: – TimeTableStatusLockComment, в котором будет содержаться следующая информация: "На данный момент бирка редактируется оператором и может быть занята. Время возможного высвобождения бирки и доступности записи на нее 2017-12-23 21:59:34.260000" (т.е. в конце комментария выводится время из TimeTableLock_lockTime + 15 мин)
--	--

10 Описание методов работы с человеком

10.1 Получение списка физических лиц, по ключевым параметрам

GET api/PersonList

Метод предназначен для получения списка пациентов по ключевым параметрам (Фамилия, Имя, Отчество, Дата рождения, СНИЛС, Серия и номер полиса).

Входящие параметры	– PersonSurName_SurName (T[50], O) – фамилия; – PersonFirName_FirName (T[50], H) – имя; – PersonSecName_SecName (T[50], H) – отчество; – PersonBirthDay_BirthDay (D, O) – дата рождения; – PersonSnils_Snils (N[11],H) – СНИЛС; – Polis_Ser (T[10], H) – серия полиса; – Polis_Num (N[15], H) – номер полиса
Ответ	Успешный ответ – список пациентов с указанием следующих параметров: – Person_id (N, O) – идентификатор пациента; – PersonSurName_SurName (T[50], O) – фамилия; – PersonFirName_FirName (T[50], H) – имя; – PersonSecName_SecName (T[50], H) – отчество; – PersonBirthDay_BirthDay (D, O) – дата рождения; – PersonSnils_Snils (N[11],H) – СНИЛС; – Polis_id (N, H) – полис: – PolisType_id (N, O) – тип полиса; – Polis_Ser (T[10], H) – серия полиса; – Polis_Num (N[15], O) – номер полиса

10.2 Получение информации по человеку

GET api/Person

Метод предназначен для получения информации по пациенту.

Входящие параметры	– Person_id (N, H) – идентификатор пациента; – PersonSurName_SurName (T[50], H) – фамилия; – PersonFirName_FirName (T[50], H) – имя; – PersonSecName_SecName (T[50], H) – отчество; – PersonBirthDay_BirthDay (D, H) – дата рождения; – PersonSnils_Snils (N[11], H) – СНИЛС; – Polis_Ser (T[10], H) – серия полиса; – Polis_Num (N[15], H) – номер полиса; – Lpu_id (N,H) – идентификатор медицинской организации (МО). Если параметр не указан, то метод вернёт данные для всех медицинских организаций текущего региона; – KLCountry_id (N,H) – идентификатор страны гражданства; – Document_IsTwoNation (N,H) – признак "Двойное гражданство";
---------------------------	--

	параметр принимает значения "0" – нет, "1" – да; – allFields (boolean, H) – все поля. Если передан параметр allFields или ResponseFull, то в ответ выводятся вся информация по человеку; – ResponseFull (boolean, H) – полный ответ. Если передан параметр allFields или ResponseFull, то в ответ выводятся вся информация по человеку; – OtherRegion_MoAttach (boolean, H) – пациенты, прикрепленные к МО другого региона
Ответ	При запуске метода проводится проверка принадлежности пользователя к учётной записи типа "Вендор API". В случае положительного ответа принадлежность пользователя к МО не проверяется. – Если не передан ни один параметр, то ошибка: – код состояния: 200 – значение параметров: – error_code:6 – error_msg:"В запросе должен быть хотя бы один параметр." – Если не найдено ни одной записи: – код состояния: 200 – значение параметров: – error_code: 0 – data: [] – Если при обработке запроса обнаружено, что в параметре Polis_Num введено значение, не удовлетворяющее контролю контрольного разряда: – код состояния: 200; – значение параметров: – error_code:6; – error_msg:"Номер ЕНП заполнен не верно". – Если хотя бы одна запись найдена, то успешный ответ – список людей с информацией: – Person_id (N, O) – идентификатор пациента; – PersonSurName_SurName (T[50], O) – фамилия; – PersonFirName_FirName (T[50], H) – имя; – PersonSecName_SecName (T[50], H) – отчество; – PersonBirthDay_BirthDay (D, H) – дата рождения; – Person_Sex_id (N, H) – пол; – PersonPhone_Phone (T, H) – телефон; – PersonSnils_Snils (N[11],H) – СНИЛС; – PersonInn_Inn (N, H) – ИНН пациента; – SocStatus_id (N, H) – социальный статус; – SocStatusFed_code (T[20]) – социальный статус по коду справочника Минздрава; – Person_pid (N, H) – идентификатор представителя пациента; – ParPersonSecName_SecName (T[50]) – отчество представителя пациента; – DeputyKind_id (N, H) – статус представителя; – UAddress_id (N, H) – адрес регистрации; – UAddress_Address (T[300]) – адрес регистрации строкой; – PAddress_id (N, H) – адрес проживания; – PAddress_Address (T[300]) – адрес проживания строкой; – Org_id (N, H) – место работы; – BAddress_id (N, H) – адрес рождения; – Org_id (N, H) – место работы; – Post_id (N, H) – должность;

	– BDZ_guid (N, H) – GUID по БДЗ; – BDZ_id (N, H) – идентификатор по БДЗ; – LegalStatusVZN_id (N) – идентификатор правового статуса нерезидента – LegalStatusVZN_Name (T) – наименование правового статуса нерезидента; – LegalStatusVZN_pid (N) – идентификатор базового правового статуса нерезидента; – LegalStatusVZN_pName (T) – наименование базового правового статуса нерезидента; – Lpu_id (N) – идентификатор медицинской организации прикрепления. – KLCountry_id (N,H) – идентификатор страны гражданства; – Document_IsTwoNation (N,H) – признак "Двойное гражданство"; параметр принимает значения "0" – нет, "1" – да; – Employment_id (N, H) – идентификатор занятости пациента; – FamilyStatus_id (N,H) – идентификатор семейного положения пациента; – PersonFamilyStatus_IsMarried (N,H) – идентификатор типа нахождения в зарегистрированном браке; – DocumentAuthority_id (N,H) – идентификатор типа документа законного представителя; – DocumentDeputy_Ser (T, H) – серия документа законного представителя; – DocumentDeputy_Num (T, H) – номер документа законного представителя; – DocumentDeputy_Issue (T, H) – кем выдан документ законного представителя; – DocumentDeputy_begDate (D,H) – дата выдачи документа законного представителя; – deputyOrg_id (N, H) – идентификатор организации, являющейся законным представителем пациента
--	---

10.3 Создание человека

POST api/Person

Метод предназначен для создания человека.

Алгоритм выполнения метода:

а) Если признак "Личность неизвестна" (Person_isUnknown):

- НЕ заполнен или "Нет", то обязательными параметрами являются: Фамилия (PersonSurName_SurName), Дата рождения (PersonBirthDay_BirthDay), Пол (Person_Sex_id), Социальный статус (SocStatus_id). Если не все обязательные параметры заполнены, то в ответе возвращается ошибка "Отсутствует один из обязательных параметров PersonSurName_SurName,

PersonBirthDay_BirthDay, Person_Sex_id, SocStatus_id". Дальнейшие действия НЕ производятся.

- "Да", то обязательным параметром является Фамилия (PersonSurName_SurName). Если не все обязательные параметры заполнены, то в ответе возвращается ошибка "отсутствует параметр PersonSurName_SurName". Дальнейшие действия НЕ производятся.
- б) Проводится проверка на дублирование пациента: если в БД есть человек с такими же параметрами, которые указаны во входных параметрах метода, то ошибка "Данные человека не прошли проверку на дублирование". Новый человек в БД не добавляется.
- в) Проводится проверка: параметр PersonPhone_Phone может содержать только 10 значное число. Если результат проверки отрицательный, то выводится ошибка о неверном формате входного параметра;
- г) Для региона Базовый: Если в любом передаваемом параметре передается значение, не соответствующее ожидаемому формату, например:
 - ожидается числовое значение, а передано текстовое значение;
 - ожидается параметр, состоящий из 11 символов, а передан параметр, состоящий из большего или меньшего количества символов;
 то, ошибка: "Неверный формат поля <Наименования полей через запятую>". Создание человека в БД не происходит.
- д) Для региона Базовый: Если переданная дата рождения больше текущей даты, то ошибка "Некорректное значение в поле PersonBirthDay_BirthDay". Создание человека не происходит.
- е) Для региона Базовый: Проверка на корректность и обязательность поля СНИЛС:

Обязательность заполнения поля "СНИЛС" зависит от установленных настроек. Если на форме "Параметры системы" на уровне "Контроль на обязательность полей" в разделе "Человек" в поле "Обязательность заполнения поля "СНИЛС" установлено значение "Запрет сохранения", то поле "СНИЛС" обязательно, если установлено значение "Отключен" или "Предупреждение", то поле необязательно.

- Если поле СНИЛС является обязательным, но в методе передано пустое значение параметра СНИЛС, то ошибка: "Отсутствует один из обязательных параметров PersonSurName_SurName, PersonBirthDay_BirthDay,

Person_Sex_id, SocStatus_id, PersonSnils_Snils". Новый человек в БД не добавляется.

- Проводится проверка: параметр СНИЛС может содержать только 11значный номер, состоящий из цифр. Если результат проверки отрицательный, то выводится ошибка о неверном формате входного параметра;
- ж) Для региона Базовый: Если в передаваемом параметре (Person_Sex_id, SocStatus_id, Address_id, PAddress_id, BAddress_id, Org_id, Post_id) передается значение id, не найденное в системе, ошибка: "Значения для <Наименования полей через запятую> не найдены в справочниках системы", создание человека в БД не происходит. Аналогичный контроль производится для параметров, по которым ожидается значение 0/1 (да/нет) (Person_isUnknown). Если контроль не пройден, то ошибка "Значение поля <названия полей через запятую> не соответствует ожидаемому", создание человека в БД не происходит.
- з) Для региона Базовый: Если в общих настройках параметров системы в Контроле на обязательность полей для "Контроля на корректность ИНН" установлено значение "Запрет сохранения", то в случае, если в поле ИНН было передано непустое значение, проводится проверка на контрольную сумму. Если проверка не пройдена, создание Человека не происходит, ошибка "Ошибка проверки контрольной суммы в ИНН".
- и) Правила для параметров:
 - PersonSurName_SurName (T, O): Если параметр указан и пустой, то метод вернёт ошибку: Значения для (PersonSurName_SurName) обязательное и не может быть пустым.
 - PersonBirthDay_BirthDay (D, O): Если параметр не указан, то он не проверяется. Если параметр указан и пустой, то метод вернёт ошибку: Значения для (PersonBirthDay_BirthDay) обязательное и не может быть пустым.
 - Person_Sex_id (N, O): Если параметр не указан, то он не проверяется. Если параметр указан и пустой, то метод вернёт ошибку: Значения для (Person_Sex_id) обязательное и не может быть пустым.
 - SocStatus_id (N, O): Если параметр не указан, то он не проверяется. Если параметр указан и пустой, то метод вернёт ошибку: Значения для (SocStatus_id) обязательное и не может быть пустым.

Входящие	– Person_isUnknown (N, H) – признак "Личность неизвестна";
-----------------	--

параметры	– PersonSurName_SurName (Т, О) – фамилия; – PersonFirName_FirName (Т, Н) – имя; – PersonSecName_SecName (Т, Н) – отчество; – PersonBirthDay_BirthDay (D, О) – дата рождения; – Person_Sex_id (N, О) – пол; – PersonPhone_Phone (N, Н) – телефон; – PersonSnils_Snils (N,Н) – СНИЛС; – SocStatus_id (N, О) – социальный статус; – UAddress_id (N, Н) – адрес регистрации; – PAddress_id (N, Н) – адрес проживания; – BAddress_id (N, Н) – адрес рождения; – Org_id (N, Н) – место работы; – Post_id (N, Н) – должность; – PersonInn_Inn (N, Н) – ИНН; – KLCountry_id (N,Н) – идентификатор страны гражданства; – Document_IsTwoNation (N,Н) – признак "Двойное гражданство"; параметр принимает значения "0" – нет, "1" – да; – Person_pid (N, У) – идентификатор представителя несовершеннолетнего; если DeputyKind_id не равен "1" ("0", "Отсутствует"), тогда параметр обязателен для заполнения; – DeputyKind_id (N, У) – идентификатор типа попечителя; обязателен для заполнения, если указан Person_pid; – Employment_id (N, Н) – идентификатор занятости пациента; – FamilyStatus_id (N,Н) – идентификатор семейного положения пациента; – PersonFamilyStatus_IsMarried (N,Н) – идентификатор типа нахождения в зарегистрированном браке; – DocumentAuthority_id (N,Н) – идентификатор типа документа законно представителя; – DocumentDeputy_Ser (Т, Н) – серия документа законного представителя; – DocumentDeputy_Num (Т, Н) – номер документа законного представителя; – DocumentDeputy_Issue (Т, Н) – кем выдан документ законного представителя; – DocumentDeputy_begDate (D, Н) – дата выдачи документа законного представителя; – fromMobile (N,Н) – признак мобильного телефона; – options (S, Н) – настройка работы метода. Допустимые форматы: JSON или массива. Параметры настройки: – CheckChecksumSnilsOff (N, Н): – 1 – отключить проверку контрольной суммы СНИЛС; – 0 или пропущен – проверять контрольную сумму СНИЛС. – CheckDuplicateSnilsOff (N,Н): – 1 – отключить проверку дублирования СНИЛС; – 0 или пропущен – проверять СНИЛС на дублирование – DuplicateSearch (Т, Н) – список дополнительных параметров для поиска "двойников". Допустимый формат: JSON, список. Возможно указать параметры: PersonSnils_Snils, Polis_EdNum, Polis_Ser, Polis_Num, Person_Sex_id, UAddress_id; – OmsSprTerr_id (N,Н) – территория страхования; – OrgSMO_id (N,Н) – полис выдан; – PolisType_id (N,Н) – тип полиса; – PolisFormType_id (N,Н) – форма полиса; – Polis_EdNum (Т, Н) – ед. номер полиса; – Polis_begDate (DT, N) – дата выдачи полиса;
-----------	--

	– Polis_endDate (DT, N) – дата закрытия полиса; – Polis_Ser (T, H) – серия полиса; – Polis_Num (T, H) – номер полиса
Ответ	В случае корректной обработки запроса и успешного создания записи в ответе будет прислано: – код статуса: 200 – значение параметров: – error_code: 0 – data: – Person_id: значение параметра Если в запросе не заполнен хотя бы один из обязательных параметров, то: – код статуса: 200 – значение параметров: – error_code: 6 – error_msg: "Отсутствует один из обязательных параметров PersonSurName_SurName, PersonBirthDay_BirthDay, Person_Sex_id, SocStatus_id" Регион: Базовый Если в параметре SocStatus_id указано значение "99", то сохранение значения социального статуса осуществляется в зависимости от возраста человека на текущую дату (возраст человека определяется по значению во входном параметре PersonBirthDay_BirthDay запроса: – <=7; – >= возраст на текущую дату <= 18; – >18 возраст на текущую дату <=60; – возраст на текущую дату > 60;
Описание ошибок	– признак "Личность неизвестна" (Person_isUnknown) не заполнен или его значение "Нет", и заполнены не все обязательные параметры – отображается ошибка "Отсутствует один из обязательных параметров PersonSurName_SurName, PersonBirthDay_BirthDay, Person_Sex_id, SocStatus_id"; – значение признака "Личность неизвестна" (Person_isUnknown) "Да" и заполнены не все обязательные параметры – отображается ошибка "Отсутствует параметр PersonSurName_SurName"; – в БД есть человек с такими же параметрами, которые указаны во входных параметрах метода – отображается ошибка "Человек с такими данными уже присутствует в базе"; – параметр PersonPhone_Phone содержит не десятизначное число – отображается ошибка "Неверный формат телефонного номера. Телефонный номер должен состоять из 10 цифр"; – в любом передаваемом параметре передается значение, не соответствующее ожидаемому формату – отображается ошибка "Неверный формат поля <Наименования полей через запятую>"; – ожидается параметр, состоящий из 11 символов, а передан параметр, состоящий из большего или меньшего количества символов – отображается ошибка "Номер СНИЛС должен содержать 11 символов"; – переданная дата рождения больше текущей даты – отображается ошибка

	"Некорректное значение в поле PersonBirthDay_BirthDay"; – в передаваемом параметре (Person_Sex_id, SocStatus_id, Address_id, PAddress_id, BAddress_id, Org_id, Post_id) передается значение id, не найденное в Системе – отображается ошибка "Значения для <Наименования полей через запятую> не найдены в справочниках Системы"; – значение параметров, по которым ожидается значение 0/1 (да/нет) (Person_isUnknown), не найденное в Системе – отображается ошибка "Значение поля <Названия полей через запятую> не соответствует ожидаемому"; – в общих настройках параметров Системы в Контроле на обязательность полей для "Контроля на корректность ИНН" установлено значение "Запрет сохранения", и в поле ИНН было передано непустое значение, проводится проверка на контрольную сумму – отображается ошибка "Ошибка проверки контрольной суммы в ИНН"; – включена проверка контрольной суммы СНИЛС, и она не пройдена – отображается ошибка "Ошибка проверки контрольной суммы в СНИЛС"; – включена проверка СНИЛС на дублирование, и она не пройдена – отображается ошибка "Человек с введённым номером СНИЛС уже есть в базе"
--	--

10.4 Редактирование данных человека

PUT api/Person

Метод предназначен для редактирования информации по человеку.

Входящие параметры	– Person_id (N, O) – идентификатор пациента; – PersonSurName_SurName (S[50], H) – фамилия; – PersonFirName_FirName (S[50], H) – имя; – PersonSecName_SecName (S[50], H) – отчество; – PersonBirthDay_BirthDay (D, H) – дата рождения; – Person_Sex_id (N, H) – пол; – PersonPhone_Phone (N[10], H) – телефон; – PersonSnils_Snils (N[11], H) – СНИЛС; – SocStatus_id (N, H) – социальный статус; – UAddress_id (N, H) – адрес регистрации; – PAddress_id (N, H) – адрес проживания; – BAddress_id (N, H) – адрес рождения; – Org_id (N, H) – место работы; – Post_id (N, H) – должность; – PersonInn_Inn (N, H) – ИНН; – KLCountry_id (N, H) – идентификатор страны гражданства; – Document_IsTwoNation (N, H) – признак "Двойное гражданство"; параметр принимает значения "0" – нет, "1" – да; – Person_pid (N, Y) – идентификатор представителя несовершеннолетнего; если DeputyKind_id не равен "1" ("0", "Отсутствует"), тогда параметр обязателен для заполнения; – DeputyKind_id (N, Y) – идентификатор типа попечителя; обязателен для заполнения, если указан Person_pid; – Employment_id (N, H) – идентификатор занятости пациента;
---------------------------	--

	– FamilyStatus_id (N,H) – идентификатор семейного положения пациента; – PersonFamilyStatus_IsMarried (N,H) – идентификатор типа нахождения в зарегистрированном браке; – DocumentAuthority_id (N,H) – идентификатор типа документа законного представителя; – DocumentDeputy_Ser (T, H) – серия документа законного представителя; – DocumentDeputy_Num (T, H) – номер документа законного представителя; – DocumentDeputy_Issue (T, H) – кем выдан документ законного представителя; – DocumentDeputy_begDate (D,H) – дата выдачи документа законного представителя; – DeputyOrg_id (N,H) – идентификатор организации-попечителя; – CheckDeputyTies (N,H) – признак: контролировать допустимость типа представителя (Если контроль включён и тип недопустим, то метод вернёт ошибку); – options (S, H) – настройка работы метода. Допустимые форматы: JSON или массива. Параметры настройки: – CheckChecksumSnilsOff (N, H): – 1 – отключить проверку контрольной суммы СНИЛС; – 0 или пропущен – проверять контрольную сумму СНИЛС; – CheckDuplicateSnilsOff (N,H): – 1 – отключить проверку дублирования СНИЛС; – 0 или пропущен – проверять СНИЛС на дублирование; – DuplicateSearch (T,H) – список дополнительных параметров для поиска "двойников". Допустимы форматы: JSON, список. Допустимые параметры: PersonSnils_Snils, PersonInn_Inn, Polis_EdNum. При передаче пустых параметров можно очистить все поля пациента, кроме PersonSurName_SurName, PersonBirthDay_BirthDay, Person_Sex_id, SocStatus_id. Текущие правила: – Нельзя указать одновременно Person_pid и DeputyOrg_id. Один из них должен быть пустым или не указан. – Если параметр DeputyKind_id указан и пустой – запись об организации-попечителе удаляется. – Если параметр DeputyKind_id указан и =1 (тип "Отсутствует") – информация об организации-попечителе удаляется. При этом параметр DeputyOrg_id игнорируется. – Если указан только DeputyOrg_id и он пустой – текущая запись об организации-попечителе удаляется. – Если указан DeputyOrg_id и он не пустой – в текущей записи о представителе записывается информация об организации-попечителе. Одновременно удаляется информация о человеке-представителе (согласно правилу 1). – Если представитель добавляется (т.е. представителя ещё нет у пациента) – параметр DeputyKind_id становится обязательным
Ответ	– Проводится проверка: параметр PersonPhone_Phone может содержать только

	10–значное число. Если результат проверки отрицательный, то выводится ошибка о неверном формате входного параметра; – Если по переданному значению <code>Person_id</code> не найден пациент, то ошибка "Пациент не найден в системе". – Если в любом передаваемом параметре передается значение, не соответствующее ожидаемому формату, например: – ожидается числовое значение, а передано текстовое значение; – ожидается параметр, состоящий из 11 символов, а передан параметр, состоящий из большего или меньшего количества символов; то ошибка: "Неверный формат поля <Наименования полей через запятую>". Обновление человека в БД не происходит. – Проводится проверка на дублирование пациента: если в БД есть человек с такими же параметрами, которые указаны во входных параметрах метода (т.е. если для имеющегося человека <code>Person_id</code> передана Фамилия, по которой вместе с другими обязательными параметрами проверки на дубль будет найден другой Человек с совпадающими параметрами), то ошибка "Данные человека не прошли проверку на дублирование". Данные по человеку в БД не обновляются. – Если в любом передаваемом параметре передается значение <code>id</code> (<code>Person_Sex_id</code>, <code>SocStatus_id</code>, <code>Address_id</code>, <code>PAddress_id</code>, <code>BAddress_id</code>, <code>Org_id</code>, <code>Post_id</code>), не найденное в системе, ошибка: "Значения для <Наименования полей через запятую> не найдены в справочниках системы", обновление данных человека в БД не происходит. Аналогичный контроль производится для параметров, по которым ожидается значение 0/1 (да/нет) (<code>Person_isUnknown</code>). Если контроль не пройден, то ошибка "Значение поля <названия полей через запятую> не соответствует ожидаемому", обновление данных человека в БД не происходит. – Производится проверка: параметр СНИЛС может содержать только 11–тизначное значение (при проверке исключаются дефисы, которые могут быть переданы в параметре <code>SocStatus_id</code>), состоящее из цифр. Если результат проверки отрицательный, то выводится ошибка о неверном формате входного параметра. – Правила для параметров: – <code>PersonSurName_SurName (T, Y): ...</code> Если параметр не указан, то он не проверяется. Если параметр указан и пустой, то метод вернёт ошибку: Значения для (<code>PersonSurName_SurName</code>) обязательное и не может быть пустым. – <code>PersonBirthDay_BirthDay (D, Y): ...</code> Если параметр не указан, то он не проверяется. Если параметр указан и пустой, то метод вернёт ошибку: Значения для (<code>PersonBirthDay_BirthDay</code>) обязательное и не может быть пустым. – <code>Person_Sex_id (N, Y): ...</code> Если параметр не указан, то он не проверяется.
--	---

	Если параметр указан и пустой, то метод вернёт ошибку: Значения для (Person_Sex_id) обязательное и не может быть пустым.. – SocStatus_id (N, Y): ... Если параметр не указан, то он не проверяется. Если параметр указан и пустой, то метод вернёт ошибку: Значения для (SocStatus_id) обязательное и не может быть пустым. Если запрос обработан корректно, и данные успешно обновлены, то: – код состояния: 200; – значение параметров: – success: true; – error_code: 0; – data: – Person_id: значение параметра. Если в запросе не указан обязательный параметр Person_id, то: – код состояния: 200; – значение параметров: – error_code: 6; – error_msg: Отсутствует обязательный параметр 'Person_id'. Если в запросе указан неверный формат номера: – код состояния: 200; – значение параметров: – success: false; – error_code: 6; – error_msg: Неверный формат телефонного номера. Телефонный номер должен состоять из 10 цифр
--	---

10.5 Получение записей на прием к врачу

GET api/TimeTableGraf/TimeTableGrafByMedStaffFact

Входящие параметры	– MedStaffFact_id (N, O) – место работы врача – TimeTableGraf_beg (DT, O) – дата и время начала диапазона – TimeTableGraf_end (DT, O) – дата и время окончания диапазона
Ответ	Успешный ответ – список бирок и пациентов, записанных к врачу в указанный диапазон: – TimeTableGraf_id (N, O) – идентификатор бирки – TimeTableGraf_begTime (DT, O) – дата и время приема – TimeTableGraf_factTime (DT, H) – фактическое время приема. – Person_id (N, O) – идентификатор пациента

10.6 Получение прикреплений пациента

GET api/PersonAttach

Входящие параметры	- Person_id (N, O) – идентификатор пациента; - Lpu_id (N, H) – идентификатор МО; - LpuRegion_id (N, H) – идентификатор участка; - LpuAttachType_id (N, H) – идентификатор типа прикрепления
Ответ	Ошибки: Если person_id не найден, то метод возвращает "error_code", равный 0. Если не заполнен обязательный параметр person_id, то метод возвращает "error_code", равный 6, и сообщение об ошибке: "Отсутствует обязательный параметр 'Person_id'". Успешный ответ: - PersonSnils_Snils (N[11], H) – СНИЛС; - Org_id (N, H) – место работы; - Post_id (N, H) – должность. Список прикреплений: - PersonCard_id (N, H) – идентификатор прикрепления; - PersonCard_begDate (D, H) – дата прикрепления; - PersonCard_endDate (D, H) – дата открепления; - PersonCard_Code (N, H) – номер амбулаторной карты; - Lpu_id – идентификатор МО прикрепления; - Lpu_Name – полное наименование МО прикрепления; - Lpu_Nick – краткое наименование МО прикрепления; - LpuRegion_Name – наименование участка; - LpuRegion_id (N, H) – идентификатор участка прикрепления; - LpuAttachType_id (N, H) – идентификатор типа прикрепления; - PersonCard_isAttachAuto (N, H) – признак автоприкрепления; - CardCloseCause_id (N, H) – идентификатор причины закрытия карты; - PersonCard_AttachAutoDT (DT, H) – дата и время автоматического прикрепления; - PersonCard_isAttachCondit (N, H) – признак условного прикрепления
Пример	<pre>{ "error_code": 0, "data": [{ "PersonSnils_Snils": null, "Org_id": "1190018853", "Post_id": null, "attach_data": [{ "PersonCard_id": "77306518970", "PersonCard_begDate": "2020-04-02", "PersonCard_endDate": null, "PersonCard_Code": "943218325458", "Lpu_id": "10010833", "Lpu_Name": "ГБУЗ П КРАЯ ГОРОДСКАЯ КЛИНИЧЕСКАЯ БОЛЬНИЦА</pre>

	№2", "Lpu_Nick": "ГКП 2", "LpuRegion_Name": "777765", "LpuRegion_id": "77300002775", "LpuAttachType_id": "1", "PersonCard_isAttachAuto": null, "CardCloseCause_id": null, "PersonCard_AttachAutoDT": null, "PersonCard_isAttachCondit": "1", "LpuRegionType_id": "1", "LpuRegionType_Name": "Терапевтический" }] }] }
--	--

10.7 Получение основного прикрепления пациента

GET api/PersonMainAttach

Назначение метода	Возвращаются данные по МО действующего основного прикрепления пациента (тип участка – терапевтический или педиатрический, если пациенту меньше 18 лет), а также данные участкового врача.
Входящие параметры	Person_id (N, O) – идентификатор пациента.
Ответ	Успешный ответ: – Person_id (N, O) – идентификатор пациента; – Lpu_id (N, O) – идентификатор МО прикрепления; – LpuUnit_FRMOid (T, O) – единый уникальный идентификатор медицинской организации (OID). Порядок определение OID: – У группы отделений заполнено поле OID ФРМО (LpuUnit.FRMOid); – Иначе, у группы отделений сохранена ссылка на справочник "ФРМО. Справочник структурных подразделений"; – Отделение из группы отделений связано с записью справочника "ФРМО. Справочник отделений и кабинетов": по ней определяется связанная запись по справочнику "ФРМО. Справочник структурных подразделений"; – В случае отсутствия OID у структурного подразделения передается OID МО (используется PassportToken_tid из fed.PassportToken, передаются только OID нового образца); – Lpu_Name (T, O) – наименование МО. Порядок определения наименования: <Наименование МО> +<Наименование Группы отделений>, где Наименование Группы отделений определяется след. образом:

	– Если у группы отделений заполнено поле OID ФРМО (LpuUnit.FRMOid), то наименование группы отделений; – Наименование медицинской организации; – Наименование МО в Реестре МО РФ (fed.PassportToken); – Address_Address (Т, О) – адрес фактического местоположения структурного подразделения медицинской организации. Указывается адрес структурного подразделения медицинской организации следующем формате: почтовый индекс, регион, населенный пункт, улица, дом. Пример: 411228, Саратовская обл., г. Саратов, ул. им. Черемушкина Н. Г., д. 141; – Org_Phone (Т, Н) – контактный телефон МО (регистратура); – MedStaffFact_id (N, О) – идентификатор места работы врача. Врач определяется по участку прикрепления пациента Person_id среди открытых мест работы на участке (открытые на дату запроса, в данном случае, текущая дата запроса) с признаком «Основной врач». Если для выбранного участка нет основного врача либо его период работы закрыт или не наступил еще, то передавать первого найденного врача с участка. Алгоритм определения врачей на участке: – проверить даты начала и окончания работы врачей на участке MedStaffRegion_begDate и MedStaffRegion_endDate. Текущая дата запроса должна попадать в период работы врача на участке; – проверить признак основного врача на участке (MedStaffRegion_isMain = 2); – если на участке нет основного врача, в период которого попадает дата запроса, то выбрать первого из списка врача, у которого текущая дата запроса попадает в период работы на участке; – MedStaffFact_Fio (Т, О) – Ф. И. О. врача участка
Ошибки	Возможные ошибки: – если в запросе не заполнен входящий параметр, отобразится ошибка (прикладной код: "6") с текстом: "Отсутствует обязательный параметр 'Person_id'"; – если по Person_id не найдено действующего основного прикрепления пациента, отобразится ошибка (прикладной код: "1") с текстом: "Не найдено действующего основного прикрепления пациента!"
Пример	Успешный ответ: <pre>{ "error_code": 0, "data": { "Person_id": "100365930", "Lpu_id": "10010833", "LpuUnit_FRMOid": "1.2.643.5.1.13.13.12.2.59.5834", "Lpu_name": "ГБУЗ ПЕРМСКОГО КРАЯ \"ГОРОДСКАЯ КЛИНИЧЕСКАЯ БОЛЬНИЦА №2\"", } }</pre>

	<pre>"Address_Address": "614068, РОССИЯ, ПЕРМСКИЙ КРАЙ, Г ПЕРМЬ, ЛУНАЧАРСКОГО УЛ, д. 90, ", "Org_phone": "2657273", "MedStaffFact_id": null, "Person_Fio": null } }</pre>
--	--

11 Запись на прием к врачу

11.1 Получение должности по наименованию

GET api/PostByName

Метод предназначен для получения должности по наименованию.

Входящие параметры	– post_name (T, O) – наименование должности; – sess_id (N, O) – идентификатор сессии
Ответ	Если запрос был обработан успешно и в БД была найдена хотя бы одна запись, то в ответе будет отправлено: – error_code: 0 – data: – Post_id: – Post_name: В ответе метода должны быть отправлены все записи, по которым было установлено соответствие со значением должности из запроса

11.2 Метод получения прикреплений пациента

GET api/PersonAttach

Метод предназначен для получения прикреплений пациента.

Входящие параметры	– Person_id (N, H) – идентификатор пациента; – Lpu_id (N, H) – идентификатор МО; – LpuRegion_id (N, H) – идентификатор участка; – LpuAttachType_id (N, H) – идентификатор типа прикрепления
Ответ	– PersonSnils_Snils (N[11], H) – СНИЛС; – Org_id (N, H) – место работы (значение справочника dbo.Org); – Post_id (N, H) – должность (значение справочника dbo.Post) Список прикреплений: – PersonCard_id (N, H) – идентификатор прикрепления; – PersonCard_begDate (D, H) – дата прикрепления; – PersonCard_endDate (D, H) – дата открепления; – PersonCard_Code (N, H) – номер амбулаторной карты; – Lpu_id – идентификатор МО прикрепления; – Lpu_Name – полное наименование МО прикрепления; – Lpu_Nick – краткое наименование МО прикрепления; – LpuRegion_Name – наименование участка; – LpuRegion_id (N, H) – идентификатор участка прикрепления; – LpuAttachType_id (N, H) – идентификатор типа прикрепления; – PersonCard_isAttachAuto (N, H) – признак автоприкрепления; – CardCloseCause_id (N, H) – идентификатор причины закрытия карты. – PersonCard_AttachAutoDT (DT, H) – дата и время автоматического прикрепления;

	– PersonCard_isAttachCondit (N, H) – признак условного прикрепления
Ответ	Успешный ответ: <pre> { "error_code": 0, "data": [{ "PersonSnils_Snils": null, "Org_id": "1190018853", "Post_id": null, "attach_data": [{ "PersonCard_id": "77306518970", "PersonCard_begDate": "2020-04-02", "PersonCard_endDate": null, "PersonCard_Code": "943218325458", "Lpu_id": "10010833", "Lpu_Name": "ГБУЗ ПЕРМСКОГО КРАЯ ГОРОДСКАЯ КЛИНИЧЕСКАЯ БОЛЬНИЦА №2", "Lpu_Nick": "ГКП 2", "LpuRegion_Name": "777765", "LpuRegion_id": "77300002775", "LpuAttachType_id": "1", "PersonCard_isAttachAuto": null, "CardCloseCause_id": null, "PersonCard_AttachAutoDT": null, "PersonCard_isAttachCondit": "1", "LpuRegionType_id": "1", </pre>

	<pre> "LpuRegionType_Name": "Терапевтический" }] }] } </pre>
Ошибки	Если person_id не найден: <pre> { "error_code": 0, "data": [] } </pre> Если не заполнен обязательный параметр person_id: <pre> { "error_code": 6, "error_msg": "Отсутствует обязательный параметр 'Person_id'" } </pre>

11.3 Получение МО по идентификатору

GET api/MOById

Входящие параметры	– Lpu_id (N, O) – Идентификатор МО
Ответ	Успешный ответ – Org_Name (T, O) – Наименование медицинской организации – Org_Nick (T, O) – Краткое наименование медицинской организации

11.4 Получение основного прикрепления пациента

GET api/PersonMainAttach

Назначение метода	Возвращаются данные по МО действующего основного прикрепления пациента (тип участка – терапевтический или педиатрический, если пациенту меньше 18 лет), а также данные участкового врача.
--------------------------	---

Входящие параметры	Person_id (N, O) – идентификатор пациента.
Ответ	Успешный ответ: – Person_id (N, O) – идентификатор пациента; – Lpu_id (N, O) – идентификатор МО прикрепления; – LpuUnit_FRMOid (T, O) – единый уникальный идентификатор медицинской организации (OID). Порядок определение OID: – У группы отделений заполнено поле OID ФРМО (LpuUnit.FRMOid); – Иначе, у группы отделений сохранена ссылка на справочник "ФРМО. Справочник структурных подразделений"; – Отделение из группы отделений связано с записью справочника "ФРМО. Справочник отделений и кабинетов": по ней определяется связанная запись по справочнику "ФРМО. Справочник структурных подразделений"; – В случае отсутствия OID у структурного подразделения передается OID МО (используется PassportToken_tid из fed.PassportToken, передаются только OID нового образца); – Lpu_Name (T, O) – наименование МО. Порядок определения наименования: < Наименование МО> +<Наименование Группы отделений>, где Наименование Группы отделений определяется след. образом: – Если у группы отделений заполнено поле OID ФРМО (LpuUnit.FRMOid), то наименование группы отделений; – Наименование медицинской организации; – Наименование МО в Реестре МО РФ (fed.PassportToken); – Address_Address (T, O) – адрес фактического местоположения структурного подразделения медицинской организации. Указывается адрес структурного подразделения медицинской организации следующем формате: почтовый индекс, регион, населенный пункт, улица, дом. Пример: 411228, Саратовская обл., г. Саратов, ул. им. Черемушкина Н. Г., д. 141; – Org_Phone (T, H) – контактный телефон МО (регистратура); – MedStaffFact_id (N, O) – идентификатор места работы врача. Врач определяется по участку прикрепления пациента Person_id среди открытых мест работы на участке (открытые на дату запроса, в данном случае, текущая дата запроса) с признаком «Основной врач». Если для выбранного участка нет основного врача либо его период работы закрыт или не наступил еще, то передавать первого найденного врача с участка. Алгоритм определения врачей на участке: – проверить даты начала и окончания работы врачей на участке MedStaffRegion_begDate и MedStaffRegion_endDate. Текущая дата запроса должна попадать в период работы врача на участке; – проверить признак основного врача на участке (MedStaffRegion_isMain = 2); – если на участке нет основного врача, в период которого попадает дата запроса, то выбрать первого из списка врача, у которого текущая дата запроса попадает в период работы на участке; – MedStaffFact_Fio (T, O) – Ф. И. О. врача участка
Ошибки	Возможные ошибки: – если в запросе не заполнен входящий параметр, отобразится ошибка

	(прикладной код: "6") с текстом: "Отсутствует обязательный параметр 'Person_id'"; – если по Person_id не найдено действующего основного прикрепления пациента, отобразится ошибка (прикладной код: "1") с текстом: "Не найдено действующего основного прикрепления пациента!"
Пример	Успешный ответ: <pre>{ "error_code": 0, "data": { "Person_id": "100365930", "Lpu_id": "10010833", "LpuUnit_FRMOid": "1.2.643.5.1.13.13.12.2.59.5834", "Lpu_name": "ГБУЗ ПЕРМСКОГО КРАЯ \"/> </pre>

11.5 Получение информации о докторе

GET /api/doctors/MedStaffFact_id

Метод позволяет получить информацию о враче.

Входящие параметры	– version (T, O) – версия; – sess_id (T, O) – идентификатор сессии; – apiKey (T, O) – ключ апи; – MedStaffFact_id (T, O) – идентификатор врача; – answerAsInGetDoctorsMethod (N, H) – фильтр
Выходные параметры	– id (T) – идентификатор врача; – doctor_fio (T) – Ф. И. О врач; – profile_name (T) – название профиля врача; – profile_id (T) – идентификатор профиля; – category (T) – врачебная категория;

	– lpu_nick (T) – краткое название больниц; – unit_name (T) – название подразделения; – address (T) – адрес подразделения; – feedbacks (T) – отзывы о врачах: – count (N) – количество отзывов; – comments_count (N) – количество комментариев; – rating_avg (N) – средняя оценка; – Lpu_id (T) – идентификатор больницы; – current_main (S) – основное место работы врача: – FullName (T) – Ф. И. О. врача; – QualificationCat_Name (T) – квалификационная категория; – QualificationCategory (T) – степень квалификационной категории; – Person_id (T) – идентификатор врача; – WorkData_begDate (S) – дата начала работы; – WorkType_id (T) – идентификатор типа занимаемой должности; – Dolgnost_Name (T) – название должности; – Lpu_id (T) – идентификатор больницы; – Lpu_Name (T) – название больниц; – MedSpecOms_Name (T) – специальность врача; – MedSpecOms_Code (T) – код специальности; – LpuSection_Name (T) – название отделения; – LpuUnit_Name (T) – название подразделения ; – Address_id (T) – идентификатор адреса; – MedStaffFact_id (T) – идентификатор рабочего места врача; – MedPersonal_id (T) – идентификатор медработника; – Age (N) – возраст; – LpuUnit_Address (T) – адрес подразделения; – current_addons (S) – дополнительное место работы врача – dolgnost (T) – название должности; – educations (S) – данные об образовании: – YearOfGraduation (N) – год окончания учёбы; – EducationType_Name (T) – тип обучения; – EducationInstitution_Name (T) – название учебного заведения; – Speciality_id (T) – идентификатор специальности; – DiplomaSpeciality_Name (T) – специальность в дипломе; – regions (T) – регион; – region_streets (T) – обслуживаемые улицы; – annot (T) – примечание; – status (N) – статус. Возможные значения: – 1 – есть ближайшая свободная бирка; – 2 – нет расписания; – status_msg (T) – текст с первой ближайшей записью; – can_post_feedback (B) – возможность комментирования. Возможные значения: – true – да, – false – нет – doctorInfoList (S, H) – информация о врачах при установленном фильтре doctorInfoList: – MedStaffFact_id (S) – идентификатор места работы врача; – IsPaidRec (N) – платный врач; – MedPersonal_FullName (S) – Ф. И. О. врача; – Lpu_Nick (S) – название МО; – LpuUnit_Address (S) – адрес МО; – QualificationCat_Name (S) – квалификация врача;
--	---

	– feedback_total (N) – количество оценок; – feedback_rating (N) – рейтинг врача; – firstFreeDate (S) – дата ближайшей бирки; – ProfileSpec_Name (S) – специальность; – videochatIsOn (B) – флаг - видеочат включен ; – IsAttachedDoctor (B) – находится ли доктор в прикреплении у пациента; – rating_avg (N) – рейтинг врача; – avatar_path (S) – аватар врача; – lng (N) – координаты долготы; – lat (N) – координаты широты; – firstFreeDates (S) – дата/время бирки; – source_system (S) – источник поиска врачей: pm_main – основные, pm_paid – платные
Ответ	Успешный ответ: <pre>{ "error_code": 0, "data": { "id": "99560004533", "doctor_fio": "Иванова Ирина Леопольдовна", "profile_name": "Терапевт", "profile_id": "2", "category": "Высшая", "lpu_nick": "ПЕРМЬ ГКБ 7", "unit_name": "1.ЖЕНСКАЯ КОНСУЛЬТАЦИЯ", "address": "НИКОЛАЯ ОСТРОВСКОГО, 111", "feedbacks": { "count": null, "comments_count": null, "rating_avg": null }, "Lpu_id": "10010815", "current_main": { "FullName": "ИВАНОВА ИРИНА ЛЕОПОЛЬДОВНА", "QualificationCat_Name": "Акушерство и гинекология ", "QualificationCategory": "Высшая", "Person_id": "3726446", "WorkData_begDate": { "date": "2005-01-01 00:00:00.000000", "timezone_type": 3, "timezone": "Asia/Yekaterinburg" }, }, }, }</pre>

	<pre> "WorkData_endDate": null, "WorkType_id": "1", "Dolgnost_Name": "Врач-акушер-гинеколог", "Lpu_id": "10010815", "Lpu_Name": "МУЗ \Городская Клиническая Больница №7\\"", "MedSpecOms_Name": "Лечебное дело. Педиатрия/Акушерство и гинекология", "MedSpecOms_Code": "8", "LpuSection_Name": "Отделение (акушер гин прием жк)_13", "LpuUnit_Name": "1.ЖЕНСКАЯ КОНСУЛЬТАЦИЯ", "Address_id": "5802991", "MedStaffFact_id": "99560004533", "MedPersonal_id": "3368", "MedStaffFactCache_CostRec": null, "Age": 61, "LpuUnit_Address": "г Пермь, ул Николая Островского, д 111" }, "dolgnost": "Врач-акушер-гинеколог", "educations": [{ "YearOfGraduation": 1986, "EducationType_Name": "Высшее образование - специалитет", "EducationInstitution_Name": "Федеральное государственное бюджетное образовательное учреждение высшего образования «Пермский государственный медицинский университет имени академика Е.А. Вагнера»", "Speciality_id": "15", "DiplomaSpeciality_Name": "Врач", "AcademicMedicalDegree_Name": null, "Speciality_Code": null }], "annot": "", "status": 2, "status_msg": "Нет расписания", "can_post_feedback": false } </pre>
--	---

11.6 Получение списка специальностей в МО

GET api/MedSpecOms/MedSpecOmsByMO

Входящие параметры	– Lpu_id (N, O) – Идентификатор МО – For_Record (N, Y) – параметр для определения первой свободной даты записи. Если передан, то будет считаться количество бирок доступных для записи с типами: Обычная, Запись через инфомат, Для интернета.
Ответ	Успешный ответ – список специальностей в МО, для которых есть хотя бы одно место работы на текущую дату (MedStaffFact). – MedSpecOms_id (N, O) – Идентификатор специальности. – TimetableGraf_Count (N, Y) – Количество доступных бирок для записи (выгружается, если For_Record = 1) – Регион: Пенза MedSpecClass_id (N, O) – Идентификатор специальности по V021. – Регион: Пенза MedSpecClass_Name (T, O) – Наименование специальности по V021
Пример	Запрос: http://penza.swn.local/api/MedSpecOms/MedSpecOmsByMO?forRecord=1&sess_id=g7pb894p3tdj8s3vqtfcofmum6&Lpu_id=13003995 Ответ: "MedSpecOms_id": "3815", "TimetableGraf_Count": 0

11.7 Получение списка мест работы по специальности и МО, или по должности и МО

GET api/MedStaffFact/MedStaffFactByMO

Правила метода:

- для параметров offset, limit, page – стандартные правила分页ного просмотра;
- если указан параметр offset, то параметр page игнорируется;
- при использовании page необходимо учесть, что limit имеет значение по умолчанию, поэтому с &page=3 будет искать третью страницу с числом записей по 100 на каждой;
- возвращаемый методом список упорядочен по значению MedStaffFact_id;
- если не указан ни один параметр: offset, limit, page, то метод будет работать без учета правил

Входящие	– MedSpecOms_id (N, O) – идентификатор специальности;
-----------------	---

параметры	– Lpu_id (N, O) – идентификатор МО; – Post_id (N, O) – идентификатор должности; – MedStaffFact_IsPaidRec (N, H) – признак наличия платного приёма; – TimeTableType_id (N, H) – тип бирки; – offset (N, H) – номер начальной возвращаемой записи; – limit (N, H) – максимальное число возвращаемых записей; – page (N, H) – номер страницы
Ответ	Успешный ответ – массив объектов: – MedStaffFact_id (N, O) – место работы врача; – Person_id (N, O) – идентификатор человека по месту работы; – PersonSurName_SurName (S[50], H) – фамилия; – PersonFirName_FirName (S[50], H) – имя; – PersonSecName_SecName (S[50], H) – отчество. – RecType_id (N, H) – тип записи; – LpuUnitType_id (N, O) – тип подразделения МО; – MedStaffFact_setDate (D, O) – дата начала работы сотрудника; – MedStaffFact_disDate (D, H) – дата окончания работы сотрудника; – LpuBuilding_id (N, O) – идентификатор; – Post_id (N, H) – идентификатор должности; – Lpu_id (N, H) – идентификатор МО; – LpuSection_id (N, H) – идентификатор отделения; – TimetableGraf_Count (N, H) – количество доступных (свободных) бирок с указанным типом для определенного врача. Если параметр TimeTableType_id имеет пустое значение или значение в параметре не указано, то в параметре TimetableGraf_Count будет указано количество свободных бирок с любым типом
Пример	<pre>{ "error_code": 0, "data": [{ "MedStaffFact_id": "140101000000013", "Person_id": "101010000000060", "PersonSurName_SurName": "БАНАЛЬНЫЙ", "PersonFirName_FirName": "ФЕДОР", "PersonSecName_SecName": "ИВАНОВИЧ", "RecType_id": null, "LpuUnitType_id": "1", "MedStaffFact_setDate": "01.03.2020", "MedStaffFact_disDate": null, "LpuBuilding_id": "101010000000194" }, { "MedStaffFact_id": "140101000000014", "Person_id": "101010000000060", "PersonSurName_SurName": "БАНАЛЬНЫЙ",</pre>

	<pre> "PersonFirName_FirName": "ФЕДОР", "PersonSecName_SecName": "ИВАНОВИЧ", "RecType_id": null, "LpuUnitType_id": "2", "MedStaffFact_setDate": "01.02.2020", "MedStaffFact_disDate": null, "LpuBuilding_id": "10101000000177" }, { "MedStaffFact_id": "1401010000001993", "Person_id": "101010000000002", "PersonSurName_SurName": "БРАЧ", "PersonFirName_FirName": "ТЕСТ", "PersonSecName_SecName": "", "RecType_id": null, "LpuUnitType_id": "9", "MedStaffFact_setDate": "01.01.2021", "MedStaffFact_disDate": null, "LpuBuilding_id": "101010000000074" }] } </pre>
--	---

11.8 Получение свободных дат приема

GET api/TimeTableGraf/TimeTableGrafFreeDate

Входящие параметры	– MedStaffFact_id (N, O) – место работы врача – TimeTableGraf_beg (D, O) – дата начала диапазона – TimeTableGraf_end (D, O) – дата окончания диапазона – TimeTableGrafFreeDate (N, H) – параметр учета прошедшего времени (1 – да, 0 – нет); – TimeTableType_id (N, H) – тип бирки: – если указано значение входного параметра – ответ со списком свободных дат, на которых есть бирки с указанным типом; – если пустое значение или не указан – метод работает по старым правилам
Ответ	Успешный ответ – список свободных дат – TimeTableGraf_begTime (D, O) – даты, где есть бирки со статусом "свободная". Если TimeTableGrafFreeDate=1 и свободные бирки на текущую дату отсутствуют, дата не возвращается.

11.9 Получение свободного времени приема

GET api/TimeTableGraf/TimeTableGrafFreeTime

Входящие параметры	– MedStaffFact_id (N, O) – место работы врача; – TimeTableGraf_begTime (D, O) – свободная дата приема; – Person_id (N, H) – идентификатор пациента. Правила: – если Person_id не указан, то возвращаются все виды бирок, кроме вида "Диспансерный учёт"; – если Person_id указан, и у пациента нет открытой карты диспансерного наблюдения, то возвращаются все виды бирок, кроме вида "Диспансерный учёт"; – если Person_id указан и у пациента есть открытая карта диспансерного наблюдения, то возвращаются все виды бирок, в том числе вида "Диспансерный учёт"; – TimeTableType_id (N, H) – тип бирки; – TimeTableGrafFreeDate (N, H) – параметр учета прошедшего времени ("1" – да, "0" – нет). Если значение не указано или указано "0", то метод возвращает бирки со временем начала, большим текущего времени. Если в параметре передано значение "1", то метод возвращает все бирки за указанную дату; – offset (N, H) – номер начальной возвращаемой записи; – limit (N, H) – максимальное число возвращаемых записей. Значение по умолчанию – 10000; – page (N, H) – номер страницы. Значение по умолчанию – 1
Ответ	Успешный ответ – массив свободных бирок на прием к врачу: { – TimeTableGraf_id (N, H) – идентификатор свободной бирки. Если по пациенту (Person_id) найдена открытая контрольная карта диспансерного наблюдения в отделении МО места работы врача и специальность врача совпадает со специальностью ответственного врача, указанного в карте дисп. наблюдения, то дополнительно передаются идентификаторы свободных бирок с типом "Диспансерный учёт"; – TimeTableGraf_begTime (DT, O) – дата и время начала приема; – TimeTableGraf_Time (N,O) – длительность приема; – TimeTableType.id (N,H) – Регион Калуга. Тип бирки; – TimeTableType.Name (T,H) – Регион Калуга. Строковый тип бирки } Выходной параметр для случаев, когда бирка, на которую пытаются записаться, заблокирована: – TimeTableStatusLockComment (T, H) – сообщение о блокировке бирки. Сообщение содержит в себе возможное время разблокировки бирки. Проверка на наличие переданной во входных параметрах бирки в таблице БД по TimeTableMedService_id: – если TimeTableMedService_id есть в БД, записывать на бирку нельзя. Производится подсчет минут от даты и времени, указанных в БД исходя из логики, что бирка может быть заблокирована оператором 15 минут. Соответственно, если TimeTableLock_lockTime=2017-12-23 21:59:34.260000, то бирка станет доступной для записи или занятой в 2017-12-23 22:14:34.260000; – в связи с данной проверкой для бирок, которые нашлись в таблице БД выводится в ответе параметр: – TimeTableStatusLockComment, в котором будет содержаться следующая информация: "На данный момент бирка

	редактируется оператором и может быть занята. Время возможного высвобождения бирки и доступности записи на нее 2017-12-23 21:59:34.260000" (т.е. в конце комментария выводится время из TimeTableLock_lockTime + 15 мин)
--	--

11.10 Запись пациента на прием

POST api/TimeTableGraf/TimeTableGrafWrite

Входящие параметры	– Person_id (N, O) – идентификатор пациента; – TimeTableGraf_id (N, O) – идентификатор свободной бирки; – EvnQueue_id (N, H) – идентификатор постановки в очередь; – Регион: Калуга. recsmethodtype_id (N,H) – способ записи
Ответ	– если по переданному значению Person_id не найден пациент, то ошибка "Пациент не найден в системе". – если по переданному значению TimeTableGraf_id не найдена свободная бирка, то ошибка "Не найдена свободная бирка". – если по переданному значению EvnQueue_id не найдена запись в БД, то ошибка "Не найдена запись о постановке в очередь". – если в любом передаваемом параметре передается значение, не соответствующее ожидаемому формату, например: – ожидается числовое значение, а передано текстовое значение; – ожидается параметр, состоящий из 11 символов, а передан параметр, состоящий из большего или меньшего количества символов то, ошибка: "Неверный формат поля <Наименования полей через запятую>", запись не производится. – если запись на переданную в методе бирку для данного пациента найдена в БД, то ошибка "Человек уже записан на данную бирку", запись не производится. Выходной параметр для случаев, когда бирка, на которую пытаются записаться, заблокирована: – TimeTableStatusLockComment (T, H) – сообщение о блокировке бирки. Сообщение содержит в себе возможное время разблокировки бирки. Проверка на наличие переданной во входных параметрах бирки в таблице БД по TimeTableMedService_id: – если TimeTableMedService_id есть в БД, записывать на бирку нельзя. Производится подсчет минут от даты и времени, указанных в БД исходя из логики, что бирка может быть заблокирована оператором 15 минут. Соответственно, если TimeTableLock_lockTime=2017-12-23 21:59:34.260000, то бирка станет доступной для записи или занятой в 2017-12-23 22:14:34.260000; – в связи с данной проверкой для бирок, которые нашлись в таблице БД выводится в ответе параметр: – TimeTableStatusLockComment, в котором будет содержаться следующая информация: "На данный момент бирка редактируется оператором и может быть занята. Время возможного высвобождения бирки и доступности записи на нее 2017-12-23 21:59:34.260000" (т.е. в конце комментария выводится время из TimeTableLock_lockTime + 15 мин).

	Успешный ответ – составной идентификатор записи на прием: – Person_id (N, O), TimeTableGraf_id (N, O) – составной идентификатор записи на прием. Если во входящих параметрах указан идентификатор постановки в очередь, то в направление передается TimeTableGraf_id, а Evn_Queue_id удаляется.
--	---

12 Отмена записи на прием

12.1 Отмена записи на прием, услугу по идентификатору бирки

DELETE api/TimeTable

Входящие параметры	– TimeTable_id (N, O) – идентификатор бирки; – TimeTableSource (T, O) – в зависимости от типа бирки значения: – MedService – если бирка из TimeTableMedService, Graf – если бирка из TimeTableGraf; – FailCause (N, O) – причина отмены
Ответ	Успешный ответ: – пустой ответ с кодом ошибки "0" – бирка удалена. Error_Msg – текст ошибки: – если по переданным TimeTable_id и TimeTableSource не найдена бирка, то ошибка "Бирка не найдена в системе"; – если не введен один из параметров "Отсутствуют обязательные параметры, укажите TimeTable_id и TimeTableSource"

12.2 Изменение статуса записи на прием

PUT api/TimeTableGraf/TimeTableGrafStatus

Входящие параметры	– Person_id (N, O), TimeTableGraf_id (N, O) – составной идентификатор записи на прием. – EvnStatus_id (N,O) – идентификатор статуса направления – TimeTableGraf_id (N, O) – ссылка на бирку поликлиники
Ответ	Успешный ответ – пустой ответ с кодом ошибки "0" Если EvnStatus_id меняется на 12 или 13, то в таблице dbo.TimeTableGraf значения полей RecClass_id, Person_id, EvnDirection_id меняется на NULL.

13 Запись на диспансерное наблюдение

13.1 Получение свободных дат приема службы диспансерного наблюдения

GET api/TimeTableGraf/TimeTableGrafFreeDateDisp

Регион: Базовый (кроме Казахстана).

Входящие параметры	– Person_id (N, O) – идентификатор пациента; – Usluga_id (N, O) – идентификатор услуги, выбранной пациентом (поиск по dbo.TimeTableMedService.UslugaComplexMedService_id->dbo.UslugaComplexMedService.UslugaComplex_id->dbo.UslugaComplex.Usluga_id); – TimeTableGraf_beg (D, O) – дата начала диапазона (поиск по dbo.TimeTableMedService.TimetableMedService_begTime); – TimeTableGraf_end (D, O) – дата окончания диапазона (поиск по dbo.TimeTableMedService.TimetableMedService_begTime)
Ответ	Выходные параметры: – MedService_id (N, O) – идентификатор службы, оказывающей услугу; – TimeTableGrafList(A, O) – список дат, где есть бирки со статусом "Свободная" Состав массива: – TimeTableGraf_begTime (DT, O) – дата приема. Правила обработки запроса: – определяется участок прикрепления пациента; – происходит поиск службы, которая оказывает услугу, переданную в запросе. Тип службы должен быть "Профилактические осмотры". Порядок поиска: – в группе отделений, в которую входит участок пациента: – если служба найдена, то переход к шагу 3; – если служба не найдена, то должен быть поиск в подразделении, в которое входит участок пациента: – если служба найдена, то переход к шагу 3; – если служба не найдена, то должен быть поиск в МО, в которую входит участок пациента: • если служба найдена, то переход к шагу 3; • если служба не найдена, то в ответе должна быть отправлена ошибка: • код состояния: 200; • error_code: 09004; • error_msg: Не найдена служба, подходящая для записи пациента. – если служба найдена И есть свободные бирки в указанном в запросе диапазоне (TimeTableGraf_beg – TimeTableGraf_end), то в ответе должно быть отправлено: – код состояния: 200 – error_code: 0 – data: • MedService_id • TimeTableGrafList: • TimeTableGraf_begTime.

	– если служба найдена, НО свободных бирок в указанном в запросе диапазоне нет, то в ответе должно быть отправлено: – код состояния: 200; – error_code: 09006; – error_msg: Не найдено свободное время для записи
--	---

13.2 Получение свободного времени приёма службы диспансерного наблюдения

GET api/TimeTableGraf/TimeTableGrafFreeTimeDisp

Регион: Базовый.

Входящие параметры	– MedService_id (N, O) – идентификатор службы, оказывающей услугу; – TimeTableGraf_begTime (D, O) – дата, на которую необходимо найти свободное время приёма
Ответ	Если запрос обработан успешно, и найдена хотя бы одна запись в БД, то в ответе будет отправлено: – код состояния: 200; – error_code: 0; – data: – TimeTableGraf_id (N, H) – идентификатор свободной бирки.; – TimeTableGraf_begTime (DT, O) – Дата и время начала приема; – TimeTableGraf_Time (N,O) – Длительность приема. Если запрос обработан успешно, но ни одна запись в БД не найдена, то в ответе будет отправлено: – код состояния: 200; – error_code: 09005; – error_msg: Не найдено свободное время для записи Выходной параметр для случаев, когда бирка, на которую пытаются записаться, заблокирована: – TimeTableStatusLockComment (T, H) – сообщение о блокировке бирки. Сообщение содержит в себе возможное время разблокировки бирки. Проверка на наличие переданной во входных параметрах бирки в таблице БД по TimeTableMedService_id: – если TimeTableMedService_id есть в БД, записывать на бирку нельзя. Производится подсчет минут от даты и времени, указанных в БД исходя из логики, что бирка может быть заблокирована оператором 15 минут. Соответственно, если TimeTableLock_lockTime=2017-12-23 21:59:34.260000, то бирка станет доступной для записи или занятой в 2017-12-23 22:14:34.260000; – в связи с данной проверкой для бирок, которые нашлись в таблице БД выводится в ответе параметр: – TimeTableStatusLockComment, в котором будет содержаться следующая информация: "На данный момент бирка редактируется оператором и может быть занята. Время возможного высвобождения бирки и доступности записи на нее 2017-12-23 21:59:34.260000" (т.е. в конце комментария выводится время из TimeTableLock_lockTime + 15 мин)

13.3 Получение списка услуг диспансеризации, на которые можно оформить запись

GET api/UslugaDisp

Метод предназначен для получения списка услуг диспансеризации, на которые можно оформить запись.

Регион: Базовый (кроме Казахстана).

Входящие параметры	Person_id (N, O) – идентификатор пациента
Выходные параметры	- Usluga_id (N, O) – идентификатор услуги; - Usluga_Name (S, O) – наименование услуги
Ответ	Правила формирования ответа: - в ответе должны быть только те услуги, которые: - находятся в открытом случае диспансеризации; - не были оказаны ранее, вне случая диспансеризации; - не были оказаны в самом случае диспансеризации; - есть в согласии в данном случае диспансеризации. Если у пациента есть несколько открытых случаев диспансеризации, то в ответе должны быть переданы все осмотры и исследования, на которые можно оформить запись. Если в случаях есть несколько одинаковых осмотров или исследований, то в ответе необходимо передавать только те, которые относятся к тому случаю диспансеризации, который был создан последним. Если запрос обработан успешно, и найдена хотя бы одна запись в БД, то в ответе будет: - error_code: 0 - data: - UslugaComplex_id: - UslugaComplex_name Если запрос обработан успешно, но ни одна запись в БД не найдена, то в ответе будет: - error_code: 0 - error_msg: По запросу записей не найдено

14 Обзвон пациентов записанных на прием

14.1 Получение списка прошедших и предстоящих занятых бирок в МО за заданный временной промежуток

GET api/TimeTableList

Метод предназначен для получения списка прошедших и предстоящих занятых бирок в МО за заданный временной промежуток. Поиск визитов производится в рамках МО вызова метода.

Метод разработан для получения списка занятых бирок с информацией о том, кто записан на бирку, контакт пациента с целью обзвона для подтверждения записи/визита пациента в МО.

Входящие параметры	– Lpu_id (N, H) – идентификатор медицинской организации; – TimeTable_beg (DT, H) – дата и время начала диапазона поиска (если не указано, то считать текущие время и дату за дату начала диапазона); – MedStaffFact_id (N, H) – идентификатор места работы. Правила передачи: – если передано значение и в параметре TimeTableSource передано значение "MedService", то метод вернёт бирки в соответствии с указанным типом (на службу), переданные значения параметра MedStaffFact_id будут проигнорированы; – если передано значение и при этом параметр TimeTableSource не передан, то метод вернёт все бирки с типом "на службу" и те бирки с типом "к врачу", которые соответствуют переданным значениям параметра MedStaffFact_id; – Post_id (N, H) – идентификатор должности. Правила передачи: – если передано значение и в параметре TimeTableSource передано значение "MedService", то метод вернёт бирки в соответствии с указанным типом (на службу), переданные значения параметра Post_id будут проигнорированы; – если передано значение и при этом TimeTableSource не передан, то метод вернёт все бирки с типом "на службу" и те бирки с типом "к врачу", которые соответствуют переданным в параметре Post_id. – TimeTable_end (DT,H) – дата и время окончания диапазона; – TimeTableSource (T, H) – тип бирки. Возможные значения: Graf/MedService для получения бирок на прием к врачу/на услугу службы соответственно; – offset (N,H,0) – номер начальной возвращаемой записи; – limit (N,H,100) – максимальное число возвращаемых записей; – Page (N,H,1) – номер страницы.
Ответ	Успешный ответ – список бирок, на которые записаны пациенты, в заданной МО за заданный диапазон дат. Если у УЗ пользователя есть группа "Вендор API" И в запросе была указана МО (параметр Lpu_id), то в ответе будут переданы записи только по выбранной МО. Если у УЗ пользователя есть группа "Вендор API" И в запросе не была указана

	МО (параметр Lpu_id), то в параметре ответа "data" будут переданы записи по всему региону. Если у УЗ пользователя нет группы "Вендор API", то в ответе будут переданы записи только по МО пользователя. Бирки в ответе выводить по степени близости приема к TimeTable_beg. Возвращаемые параметры: – Lpu_id (N, O) – идентификатор МО; – Org_Name (T, O) – Наименование МО; – TimeTables(S, O) – объект: – Person_id (N, O) – идентификатор пациента; – Person_SurName (T, O) – фамилия пациента; – Person_FirName (T, O) – имя пациента; – Person_SecName (T, O) – отчество пациента; – Person_Phone (T, O) – номер телефона пациента; – TimeTableSource (T, O) – тип бирки:Graf или MedService; – TimeTable_id (N, O) – идентификатор бирки; – TimeTableType_id (N, O) – тип бирки; – TimeTable_factTime (DT, H) – фактические дата и время приема для обслуженных бирок (если у бирки есть Фактическое время приема TimeTableGraf_factTime/TimeTableMedService_factTime, то пациент приходил); – TimeTable_begTime (DT, O) – дата и время начала приёма; – MedPersonal_id (N) – идентификатор сотрудника; – MedPersonal_SurName (T) – фамилия сотрудника; – MedPersonal_FirName (T) – имя сотрудника; – MedPersonal_SecName (T) – отчество сотрудника; – MedPersonal_FIO (T) – Ф. И. О. сотрудника в нормализованном виде. – MedService_id (N, H) – идентификатор службы (выводить только если тип бирки MedService); – MedService_Name (T, H) – наименование службы (выводить только если тип бирки MedService); – MedStaffFact_id (N, H) – идентификатор врача (выводить только если бирка поликлиническая Graf); – Post_id (N, H) – идентификатор должности врача (выводить только если бирка поликлиническая Graf); – Post_name (T, H) – наименование должности врача (выводить только если бирка поликлиническая Graf); – LpuSectionProfile_id (N, O) – идентификатор профиля отделения; – Usluga_Code (T, H) – код услуги; – Usluga_Name (T, H) – наименование услуги.
Возможные ошибки	– код "1", прикладной код "6", текст ошибки: "Неверное значение TimeTableSource. Должно быть Graf или MedService "; – код "2", прикладной код "6", текст ошибки: "Значение MedStaffFact_id или Post_id не найдено в системе"

14.2 Получение списка предстоящих приемов пациента в МО

GET api/TimeTableListbyPatient

Регион: Базовый (кроме Казахстана).

Метод предназначен для получения списка прошедших и предстоящих визитов пациента в МО за заданный временной промежуток. Поиск визитов производится в рамках МО вызова метода.

Входящие параметры	– Person_id (N,O) – идентификатор пациента; – TimeTable_beg (DT,H) – дата и время начала диапазона поиска (если не указано, то считать текущие время и дату за дату начала диапазона); – TimeTable_end (DT,H) – дата и время окончания диапазона; – Lpu_id (N,H) – идентификатор медицинской организации (доступно только для вендора)
Ответ	Правила: – значение параметра TimeTable_beg не может быть ранее текущих даты и времени. Иначе метод вернёт ошибку; – если параметр TimeTable_beg не задан, то устанавливается равным текущим дате и времени; – значение параметра TimeTable_end не может быть меньше параметра TimeTable_beg. Иначе метод вернёт ошибку; – если идентификатор человека не найден в Системе, то метод вернёт ошибку. Возвращаемые параметры: – Person_id (N, O) – идентификатор пациента; – Person_SurName (T, O) – фамилия пациента; – Person_FirName (T, O) – имя пациента; – Person_SecName (T, O) – отчество пациента; – Person_Phone (T, O) – номер телефона пациента; – TimeTable (S, O) – список ячеек расписания, на которые записан человек: – Lpu_id (N, O) – идентификатор медицинской организации (доступно только для вендора); – TimeTableSource (T, O) – метка источника данных; – TimeTable_id (N, O) – идентификатор ячейки расписания; – TimeTable_factTime (DT, H) – фактические дата и время приема для обслуженных ячеек расписания (если у ячейки есть фактическое время приема TimeTableGraf_factTime/TimeTableMedService_factTime, то пациент приходил); – TimeTable_begTime (T, O) – дата и время начала приёма; – MedService_id (N, H) – идентификатор службы; – MedService_Name (T, H) – наименование службы; – MedStaffFact_id (N, H) – идентификатор места работы врача; – Post_id (N, O) – идентификатор должности врача; – Post_name (T, O) – наименование должности врача; – LpuSectionProfile_id (N, O) – идентификатор профиля отделения; – Usluga_Code (T, H) – код услуги; – Usluga_Name (T, H) – наименование услуги.

14.3 Получение списка пациентов, подходящих для вакцинации от Covid–19

GET api/CovidVaccinationInfo

Входящие	– Lpu_id (N,H) – МО ИД (параметр для Вендора); – Age_Beg (N,H) – возраст начала диапазона поиска (например: 20 лет); – Age_End (N,H) – возраст конца диапазона поиска (например: 25 лет);
-----------------	---

параметры	– Sex_SysNick (Т,О) – пол (man/woman/isex); – CovidVaccinationStatus (Т,О) – выбрать значение: переболели covid-19*/не были привиты ранее**/все***; – Limit (N,O) – количество записей на странице; – page (N,O) – номер страницы
Ответ	Список прикрепленных пациентов МО старше 18 лет, которые переболели Covid-19 больше 6 мес* назад или были вакцинированы Covid-19 от больше 6 месяцев* назад (первым этапом прививки), а также не имеют медотвода от всех прививок/от данной прививки и не имеют отказа от всех прививок/отданной прививки *период 6 месяцев в соответствии с рекомендациями Метод рекомендации_Вакцинация_от_COVID_19.docx (п.2.19) (2021. Обновления НПА)– экстренная вакцинация. При переходе на рутинную вакцинацию период необходимо будет изменить на 12 мес. в соответствии с п. 2.20. ** на данный момент метод работает только для пациентов 18+ (в дальнейшем, когда будут рекомендации по прививанию детей, необходимо будет снять данное возрастное ограничение) Для пользователя с группой прав "Пользователь API" поиск пациентов производится в рамках МО, к которой принадлежит его учетная запись. Для пользователя с группой прав "Вендор" поиск пациентов производится по всем МО региона, если он не указал во входных параметрах Lpu_id, иначе – в рамках указанного Lpu_id. – CovidVaccinationStatus (Т,О) – значение, указанное во входных параметрах, отобразится как заголовок, после которого пойдет список пациентов с данным статусом; – Persons (S,O) – список пациентов: – Person_id (N,O) – ИД пациента; – Person_SurName (Т,О) – Фамилия; – Person_FirName (Т,О) – Имя; – Person_SecName (Т,О) – Отчество; – Sex_SysNick (Т,О) – пол (man/woman/isex); – Person_Age (N,O) – возраст. полных лет; – Person_Phone (Т,О) – Телефон для связи; – VaccinationType_id (N,O) – всегда 30 (т.к. для задачи поиска бровок на вакцинацию задан признак, что если VaccinationType_id=30 то это прививка от covid-19). *переболели covid-19 – есть диагноз U07.1 COVID-19, вирус идентифицирован и/или U07.2 COVID-19, вирус не идентифицирован с датой закрытия более 6 месяцев назад от даты запуска метода (вывести пациентов, найденных в п.1 по результатам проверок 2, 3, а, б,в,г); **не были привиты ранее – пациент отсутствует в журнале вакцинации vac.vac_JournalAccount с прививкой от covid-19 (вывести всех пациентов, у которых в журнале вакцинации нет прививок vac.S_VaccineType.VaccineType_id=130 и vac.S_VaccineType.VaccineType_id=131 , но они прошли все проверки из пунктов а,б,в,г и подлежат вакцинации) а также пациент не болел COVID-19 или закончил болеть более чем 6 месяцев назад; *** все – обе вышеперечисленные категории.

15 Вызов врача на дом

15.1 Добавление вызова врача на дом

POST api/HomeVisit/HomeVisit

Входящие параметры	– Person_id (N, O) – идентификатор пациента; если пациент не найден, тогда метод вернет ошибку "Пациент с ИД [%s] не найден". – CallProfType_id (N, O) – идентификатор профиля вызова; – Address_Address (T, O) – адрес вызова; – HomeVisitCallType_id (N, O) – идентификатор типа вызова; – HomeVisit_setDT (DT, O) – дата и время вызова; – HomeVisit_Num (T, H) – номер вызова; если не указан параметр HomeVisit_Num, то метод попытается создать его; если метод не сможет создать номер, то вернёт ошибку; – MedStaffFact_id (N, H) – место работы врача; – HomeVisit_Phone (T, H) – телефон обратной связи; – HomeVisitWhoCall_id (N, O) – значение из справочника "Кто вызвал" http://fixuploader.swn.local/dbspr/index/109620575; – HomeVisit_Symptoms (T, O) – симптомы; – HomeVisit_Comment (T, H) – дополнительная информация; – HomeVisitStatus_id (N, O) – идентификатор статуса вызова; – KLStreet_id (N, O) – идентификатор улицы вызова; если указан параметр KLStreet_id, то метод будет искать МО по этому параметру; если улица с таким идентификатором не найдена, то метод вернёт ошибку; если у улицы с таким идентификатором нет GUID (т.е. кода нет в БД, не привязан), то метод вернёт ошибку; – HomeVisit_StreetGUID (T, O) – GUID улицы из справочника ФИАС; если параметр неверного формата, то метод вернет ошибку "Некорректный GUID"; если улица с таким GUID не найдена, то метод вернёт ошибку "Улица с GUID [%s] не найдена"; если улица с таким идентификатором не найдена, то метод вернет ошибку "Улица с ИД [%s] не найдена"; если у улицы с таким идентификатором нет GUID, то метод вернёт ошибку "GUID улицы с ИД [%s] не найден"; если указан HomeVisit_StreetGUID и это другой адрес, то метод вернёт ошибку "Значения HomeVisit_StreetGUID и KLStreet_id указывают на разные адреса"; – KLAdr_Actual = 0 (N, H) – фильтр актуальности данных; – HomeVisit_House (T, O) – номер дома; – HomeVisit_LpuComment (T, H) – причина отказа. Обязательное, если HomeVisitStatus = "Отказ"; – Slot_Id (N, H) – идентификатор "Виртуальной бирки" вызова на дом; – LpuRegion_cid (N, H) – участок вызова; – HomeVisit_IsQuarantine (N, H) – признак "Карантин". Возможные значения: 1 и 2, где 1 - нет, 2 – да; – Report (N, H) – код отчёта – HomeVisitSource_id (N, H) – источники данных. Если параметр передан, то его значение передается в хранимую процедуру создания вызова. Если параметр не передан, то передается значение по умолчанию (12)
---------------------------	--

Ответ	Успешный ответ Код ошибки – 0 – HomeVisit_id (N) – идентификатор вызова на дом; – HomeVisitStatus_id (N, H) – идентификатор статуса вызова. Значение из справочника dbo.HomeVisitStatus; – HomeVisitStatus_name (T, H) – наименование статуса вызова. Значение из справочника dbo.HomeVisitStatus; – Slot_Id (N, H) – идентификатор "виртуальной бирки" вызова на дом. При создании вызова на дом значение Slot_Id сохраняется в dbo.HomeVisit.HomeVisit_GUID; – HomeVisit_setDT (DT, O) – дата и время вызова; – Duration (N, H) – продолжительность приема. По умолчанию передается значение 60 (в минутах); – Report (S,H) – список причин ошибок и список причин почему участок исключён из подходящих. Алгоритм работы метода: 1) Осуществляется поиск пациента по person_id в БД Промед: а) Если найден, то переход к п. 2; б) Иначе ошибка – "Если пациент не найден, тогда метод вернет ошибку "Пациент с ИД [%s] не найден". 2) Проверяется статус человека – мертв или жив в БД Промед (Person_IsDead): а) Если жив, то переход к – – если во входных параметрах указан KLStreet_id, то переход к п. 3, иначе переход к п. 4. б) Иначе ошибка – "Невозможно создать вызов на дом на умершего человека". 3) Осуществляется поиск улицы по KLStreet_id в БД Промед (таблица dbo.KLStreet): а) Если найдена, то переход к п. 4; б) Регион: Базовый (кроме Казахстана) Если не найдена, но есть территориальная единица (страна, город, нас.пункт), у которой включен признак "Вся указанная территория", то переход к п.5 (если параметр HomeVisit_StreetGUID не заполнен, то переход к п.6) в) Иначе ошибка – "Улица с ИД [%KLStreet_id] не найдена." 4) Осуществляется поиск улицы по HomeVisit_StreetGUID в БД Промед (таблица dbo.KLStreet): а) Если найдена, то переход к п. 5; б) Регион: Базовый (кроме Казахстана) Если не найдена, но есть территориальная единица (страна, город, нас.пункт), у которой включен признак "Вся указанная территория", то переход к п.6 в) Иначе ошибка – "Улица с GUID [%HomeVisit_StreetGUID] не найдена." 5) Осуществляется проверка связанности KLStreet_id с HomeVisit_StreetGUID в БД Промед: а) Если они равны, то переход к п. 6; б) Иначе ошибки: – Если у улицы с таким ИД нет GUID, то метод вернёт ошибку "GUID улицы с ИД [%s] не найден". – Если KLStreet_id с HomeVisit_StreetGUID указывают на разные адреса, то метод вернёт ошибку "Значения HomeVisit_StreetGUID и KLStreet_id указывают на разные адреса". 6) Осуществляется поиск МО по KLStreet_id и HomeVisit_House (таблица dbo.LpuRegionStreet): а) Если МО не найдена, тогда ошибка – "Не удалось найти МО,
---------------------	--

- обслуживающую указанный адрес. МО не найдена. Ошибка БД."
- б) Если МО найдена, но отмечена как тестовая, тогда ошибка – "Не удалось найти МО, обслуживающую указанный адрес. МО помечена как тестовая."
 - в) Если МО найдена, но дата окончания раньше текущей даты, тогда ошибка – "Не удалось найти МО, обслуживающую указанный адрес. Дата окончания работы МО меньше текущей даты."
 - г) Иначе переход к п. 7.
- 7) Проверяется указан ли во входных параметрах LpuRegion_cid:
- а) Если указан, то переход к п. 8;
 - б) Иначе переход к п. 10.
- 8) Проверяется МО, обслуживающая указанный адрес (значения полей KLStreet_id и HomeVisit_House) и МО указанного LpuRegion_cid:
- а) если они не совпадают, тогда ошибка "Значение LpuRegion_cid не найдено в медицинской организации, которая обслуживает указанный адрес"
 - б) иначе переход к п. 9.
- 9) Проверяется тип указанного участка LpuRegion_cid на соответствие возрасту:
- а) Если возраст пациента больше 18 лет, тогда тип участка должен быть "терапевтический", "врач общей практики", иначе ошибка – "Указанный тип участка LpuRegion_cid не соответствует возрасту пациента"
 - б) Если возраст пациента меньше 18 лет, тогда тип участка должен быть "педиатрический", иначе ошибка – "Указанный тип участка LpuRegion_cid не соответствует возрасту пациента".я
 - в) Иначе переход к п. 12
- 10) Осуществляется поиск участка, обслуживающего указанную улицу (таблица dbo.LpuRegionStreet):
- а) Если участок не найден, тогда ошибка – "Не удалось найти участок, обслуживающий указанный адрес."
 - б) Иначе переход к п. 11;
- 11) Осуществляется поиск участка, обслуживающего указанный дом:
- а) Если участок не найден, тогда ошибка – "Не удалось найти участок, обслуживающий указанный адрес. В списке обслуживаемых нет дома [%s]"
 - б) Иначе переход к п. 12;
- 12) Среди найденных участков осуществляется поиск участка соответствующего возрасту пациента:
- а) Если возраст пациента больше 18 лет, тогда осуществляется поиск участка с типом "терапевтический", "врач общей практики", если не найдено, тогда ошибка – "Не удалось найти участок, обслуживающий указанный адрес, для пациента возрастом больше 18 лет."
 - б) Если возраст пациента меньше 18 лет, тогда осуществляется поиск участка с типом "педиатрический", если не найдено, тогда ошибка – "Не удалось найти участок, обслуживающий указанный адрес, для пациента возрастом меньше 18 лет."
 - в) Иначе переход к п. 13
- 13) Проверяется значение входного параметра HomeVisitStatus:
- а) Если статус – "Назначен врач" (HomeVisitStatus_id=6), тогда переход к п. 14 :
 - б) Иначе переход к п. 16
- 14) Проверяется указан ли во входных параметрах MedstaffFact_id:
- а) Если указан то переход к п. 15;
 - б) Иначе осуществляется поиск и назначение основного врача на участке, как врача на вызов.
- 15) Проверяются рабочие места с установленным флагом "Прием на дому" и рабочие места добавленные на участки на наличие указанного места работы врача:
- а) Если оно найдено, тогда найденный врач устанавливается, как врач на вызов и переход к п. 16.
 - б) Иначе ошибка "Указанное место работы MedstaffFact_id на участке не

найдено"

16) Проверяется наличие профиля отделения участка:

- а) Если профиль отделения отсутствует, тогда ошибка – "Не удалось найти МО, обслуживающую указанный адрес. Профиль отделения не найден. Ошибка БД."
- б) Иначе переход к п.17.

17) Проверяется статус профиля отделения – "Работает по участковому принципу":

- а) Если у профиля отделения не установлен признак "Работает по участковому принципу", тогда ошибка – "Не удалось найти МО, обслуживающую указанный адрес. Профиль отделения не помечен как [Работает по участковому принципу]." (признак "Работает по участковому принципу" устанавливается на форме "Справочник dbo.LpuSectionProfile: Просмотр записи" на форме "Справочники")
- б) Иначе переход к п.18.

18) Проверяется группа отделений к которой относится найденное отделение:

- а) Если группа отделений не найдена, тогда ошибка – "Не удалось найти МО, обслуживающую указанный адрес. Группа отделений не найдена. Ошибка БД.";
- б) Иначе переход к п.19.

19) Проверяется тип группы отделений к которому относится найденное отделение:

- а) Если тип группа отделений НЕ "Поликлиника", тогда ошибка – "Не удалось найти МО, обслуживающую указанный адрес. Тип группы отделений должен быть [Поликлиника]."
- б) Иначе переход к п.20.

20) Если статус добавляемого вызова врача на дом – "Новый" (во входных параметрах HomeVisitStatus_id=1 (Новый)) или HomeVisitStatus_id=6 (Назначен врач), тогда проверяется активный вызов для данного пациента с соответствующим статусом (HomeVisitStatus_id=1 (Новый)) HomeVisitStatus_id=6 (Назначен врач) в найденной МО на день, указанный в HomeVisit_setDT:

- а) Если вызов не найден, тогда переход к п. 22.
- б) Иначе ошибка – "Для данного пациента уже имеется необслуженный вызов врача на дом".

21) Генерируется номер вызова автоматически – плюс 1 к последнему номеру вызова:

- а) Если при генерации произошла ошибка, тогда ошибка – "Невозможно сгенерировать нумератор.", "Невозможно сгенерировать номер вызова."

Алгоритм формирования отчета о работе метода:

Для получения данного отчета необходимо вызвать метод с входным параметром Report (Код отчета) для выяснения причин, почему метод не смог создать вызов врача на дом. Отличие данного отчета от возвращаемой ошибки в том, что данный отчет содержит все условия на создание вызова врача на дом и статус по ним – пройдена данная проверка или нет.

Отчёт помогает выяснить реальную причину, почему метод исключил тот или иной участок из числа подходящих для обслуживания вызова на дом.

Это поможет найти ошибки в настройках параметров участков, врачей и прочее.

Пример отчета о работе метода:

```
{
  "error_code": 0,
  "Report": [
    [
      {
        "LpuRegion_id": 101010000000005,
        "LpuRegion_Name": "777",
        "MedStaffFact_id": "1401010000001873",
        "Description": "Врач не обслуживает участок, потому что указанное место
работы в параметре MedStaffFact_id [140101000000040] не соответствует месту работы
врача."
      }
    ]
  ],
  "data": {
    "HomeVisit_id": "1401010000000737",
    "HomeVisitStatus_id": "6",
    "HomeVisitStatus_name": "Назначен врач",
    "Slot_Id": "9147d4da-ca92-40ce-bba1-07c8d1069ac5",
    "HomeVisit_setDT": "15.01.2022 10:00",
    "Duration": "60"
  }
}
```

При указании входных параметров метода стоит обратить внимание на:

- Идентификатор пациента должен существовать;
- Пациент должен быть жив;
- Идентификатор улицы и ГУИД улицы должны существовать, если они оба передаются, то должны соответствовать;
- По указанной улице должна быть обслуживающая МО;
- По указанной улице и дому должен быть обслуживающий участок;
- Тип участка должен соответствовать возрасту пациента;
- На участке должен работать действующий врач с текущим (т.е. дата окончания работы д.б. больше текущей, в т.ч. может быть не указана) местом работы;
- Тип группы отделений, к которому относится участок должен быть "Поликлиника";
- Если указывается LpuRegion_cid, тогда он должен присутствовать в МО, обслуживающей указанный адрес;
- Если указывается LpuRegion_cid, тогда тип участка должен соответствовать возрасту пациента;
- Если указывается MedStaffFact_id, тогда он должен присутствовать на участке, обслуживающей указанный адрес или на участке LpuRegion_cid.
- Если требуется автоматическое назначение врача на вызов, тогда требуется указание входного параметра HomeVisitStatus_id=6 (Назначен врач).

Успешный ответ:

	Код ошибки – 0 – HomeVisit_id (N) – Идентификатор вызова на дом – HomeVisitStatus_id (N) – Идентификатор статуса вызова. Значение из справочника dbo.HomeVisitStatus – HomeVisitStatus_name (T) – Наименование статуса вызова. Значение из справочника dbo.HomeVisitStatus – Slot_Id (N) – Идентификатор "виртуальной бирки" вызова на дом. При создании вызова на дом значение Slot_Id сохраняется в dbo.HomeVisit.HomeVisit_GUID – HomeVisit_setDT (DT) – Дата и время вызова – Duration (N) – Продолжительность приема. По умолчанию передается значение 60 (в минутах)
Ошибки	Код ошибки – 6 – Пациент с ИД [%s] не найден. – Некорректный GUID (д.б. формата #####-####-####-#####). – GUID улицы с ИД [%s] не найден. – Улица с ИД [%s] не найдена. – Значения HomeVisit_StreetGUID и KLStreet_id указывают на разные адреса. – Улица с GUID [%s] не найдена. – Не удалось найти МО, обслуживающую указанный адрес. – Не удалось найти МО, обслуживающую указанный адрес. В списке обслуживаемых нет дома [%s]. – Не удалось найти МО, обслуживающую указанный адрес. МО не найдена. Ошибка БД. – Не удалось найти МО, обслуживающую указанный адрес. МО помечена как тестовая. – Не удалось найти МО, обслуживающую указанный адрес. Дата окончания работы МО меньше текущей даты. – Не удалось найти МО, обслуживающую указанный адрес. Профиль отделения не найден. Ошибка БД. – Не удалось найти МО, обслуживающую указанный адрес. Профиль отделения не помечен как [Работает по участковому принципу]. – Не удалось найти МО, обслуживающую указанный адрес. Группа отделений не найдена. Ошибка БД. – Не удалось найти МО, обслуживающую указанный адрес. Тип группы отделений должен быть [Поликлиника]. – Не удалось найти МО, обслуживающую указанный адрес. Место работы врача на участке закрыто. – Не удалось найти МО, обслуживающую указанный адрес. Не найден врач, обслуживающий участок. – Не удалось найти участок, обслуживающий указанный адрес. – Не удалось найти участок, обслуживающий указанный адрес. Обслуживающих участков вообще нет для указанного адреса. – Не удалось найти участок, обслуживающий указанный адрес, для пациента возрастом меньше 18 лет. – Вызов врача на дом может быть оформлен через портал в рабочие дни поликлиники %s. – Невозможно сгенерировать нумератор. – Невозможно сгенерировать номер вызова. – Невозможно создать вызов на дом на умершего человека. – Ошибка запроса к БД.

	– Указанный тип участка LpuRegion_cid не соответствует возрасту пациента. – Тип участка указанного LpuRegion_cid некорректный. – Указанное место работы MedstaffFact_id на участке не найдено.
Пример	Запрос: POST http://41484.prm.bot8.promedweb.ru/api/HomeVisit/HomeVisit?CallProfType_id=1&Address_Address=РОССИЯ, ПЕРМСКИЙ КРАЙ, ПЕРМЬ Г, УРАЛЬСКАЯ УЛ, д. 2, кв. 64&HomeVisitCallType_id=1&HomeVisit_setDT=30.03.2021 09:58&HomeVisitWhoCall_id=1&Person_id=100293729&HomeVisit_StreetGUID=68bff86d-e5cb-46e6-bc78-4dee516fbc88&HomeVisit_Symptoms=Б/Л: да&HomeVisitStatus_id=1&HomeVisit_House=2 Ответ: Ошибка <pre>{ "error_code": 0, "data": [{ "HomeVisit_id": "590101000000223" }] }</pre>

15.2 Получение адреса

GET api/Address

Входящие параметры	– Person_id (N, H) – идентификатор человека; – Address_id (N, H) – идентификатор адреса; – AddressType_id (N, H) – тип адреса (1 – Адрес регистрации, 2 – Адрес проживания, 3 – Адрес рождения);
---------------------------	--

	– Area_AOGUID (Т, Н) – GUID территории (города, населенного пункта и т.п.) по ФИАС; – Street_AOGUID (Т, Н) – GUID улицы по ФИАС
Ответ	Если не задан ни Person_id, ни Address_id, то ошибка (хотя бы один из параметров должен быть задан). Если не найдено ни одной записи, то ответ "0". Если в запросе заполнен один или оба параметра AOGUIDArea, AOGUIDStreet, но не заполнены одновременно параметры Person_id и AddressType_id, то в ответе отправляется ошибка: "Отсутствует обязательный параметр (наименование параметра)". Если хотя бы одна запись найдена, то успешный ответ – список адресов с информацией: – Person_id (N, H) – идентификатор человека; – Address_id (N, O) – идентификатор адреса; – AddressType_id (N, H) – тип адреса (1 – Адрес регистрации, 2 – Адрес проживания, 3 – Адрес рождения); – Address_Zip (S[10], H) – индекс; – KLCountry_id (N, O) – страна; – KLRgn_id (N, Y) – регион; – KLSubRgn_id (N, Y) – район; – KLCity_id (N, Y) – город; – KLTown_id (N, Y) – нас. пункт; – KLStreet_id (N, O) – улица; – Address_House (N, O) – дом; – Address_Corpus (N, H) – корпус; – Address_Flat (N, H) – квартира; – Address_Address (Т, Н) – полный адрес в текстовом формате; – Address_insDT (D, O) – дата изменения записи адреса в формате ГГГГ–ММ–ДД; – AoidArea (T[36],Y) – идентификатор адресного объекта (населенного пункта, города) по ФИАС – значение справочника KLArea, поле KLArea_AOID; – AoidStreet (T[36],Y) – идентификатор адресного объекта (улицы) по ФИАС – значение справочника KLStreet, поле KLStreet_AOID; – KLAreaType_id (N, H) – идентификатор справочника тип территории; – Address_insDatetime (DT, H) – дата и время записи об адресе; – KLCountry_name (Т, O) – наименование страны; – KLRgn_name (Т, Y) – наименование региона; – KLSubRgn_name (Т, Y) – наименование района; – KLCity_name (Т, Y) – наименование города; – KLTown_name (Т, Y) – наименование населенного пункта; – KLStreet_name (Т, Y) – наименование улицы. Регион: Базовый (кроме Казахстана) – Если в запросе заполнен один или оба параметра AOGUIDArea, AOGUIDStreet, но не заполнены одновременно параметры Person_id и AddressType_id, то в ответе должна быть отправлена ошибка: – код состояния: 200

	– значение параметров: – error_code: 23001 – error_msg: Отсутствует обязательный параметр (наименование параметра)
Пример	<pre>{ "error_code": 0, "data": [{ "Person_id": "590910000078057", "Address_id": "15510240", "AddressType_id": "1", "Address_Zip": "614107", "KLCountry_id": "643", "KLCountry_name": "РОССИЯ", "KLRgn_id": "59", "KLRgn_name": "ПЕРМСКИЙ", "KLSubRgn_id": null, "KLSubRgn_name": null, "KLCity_id": "3310", "KLCity_name": "ПЕРМЬ", "KLTown_id": null, "KLTown_name": null, "KLStreet_id": "461031", "KLStreet_name": "ТРАКТОРНАЯ", "Address_House": "56", "Address_Corpus": "", "Address_Flat": "6", "Address_Address": "614107, РОССИЯ, ПЕРМСКИЙ КРАЙ, Г ПЕРМЬ,</pre>

	ТРАКТОРНАЯ УЛ, д 56, кв 6", "Address_insDT": "2020-10-09", "AoidArea": "1696D568-7A59-48D6-A81D-B982C9C3AD0F", "AoidStreet": "8557F0BF-A01F-4411-B48F-B347312ED69F", "KLAreaType_id": "1", "AOGUIDArea": "A309E4CE-2F36-4106-B1CA-53E0F48A6D95", "AOGUIDStreet": "8CB4A0FE-1BCC-4BD6-B4A8-941DF5E8F61B", "Address_insDatetime": "2020-10-09 12:10:58" }] }
--	---

15.3 Получение списка параметров адреса

GET api/AddressSearch

Метод возвращает список параметров адреса.

Входящие параметры	– address (Т,О) – Строка адреса в формализованном виде. Формат строки: Индекс, страна, область, район, город/населённый пункт, улица, номер дома, номер корпуса, номер квартиры. Допустимы форматы: „,,,город,улица,номер дома/здания,, Указанные параметры обязательны. Ведущие/начальные запятые обязательны. – analyze (N) – Признак анализа адреса. 1 – Метод вернёт дополнительную информацию результата анализа поиска адреса.
Ответ	– Country_id (N) – идентификатор страны. – Country_Code (Т) – Код страны (ISO). – Country_Name (Т) – Наименование страны. – Area_id (N) – идентификатор области/края/республики и т.д. – Area_Name (Т) – Наименование области/края/республики и т.д. – Area_Type (Т) – Тип области/края/республики и т.д. – Area_KLADR_Code (Т) – Код КЛАДР области/края/республики и т.д. – Area_Postal_Index (Т) – Почтовый индекс области/края/республики и т.д. – Area_AOGUID (Т) – идентификатор адресного объекта по ФИАС (области/края/республики и т.д.). – SubRegion_id (N) – идентификатор района. – SubRegion_Name (Т) – Наименование района. – SubRegion_Type (Т) – Тип района.

	– SubRegion_KLADR_Code (T) : Код КЛАДР района. – SubRegion_Postal_Index (T) – Почтовый индекс района. – SubRegion_AOGUID (T) – идентификатор адресного объекта по ФИАС (района). – City_id (N) – идентификатор города. – City_Name (T) – Наименование населённого пункта (города, деревни и т.п.) – City_Type (T) – Тип населённого пункта. – City_KLADR_Code (T) – Код КЛАДР населённого пункта. – City_Postal_Index (T) – Почтовый индекс населённого пункта. – City_AOGUID (T) – идентификатор адресного объекта по ФИАС (населённого пункта). – Street_id (N) – идентификатор улицы. – Street_Name (T) – Наименование улицы (переулка, бульвара и т.п.). – Street_Type (T) – Тип улицы. – Street_KLADR_Code (T) – Код КЛАДР улицы. – Street_Postal_Index (T) – Почтовый индекс улицы. – Street_AOGUID (T) – идентификатор адресного объекта по ФИАС (улицы). – House (T) – Номер дома (строения, здания и т.п.). – House_Corpus (T) – Номер корпуса. – House_Type (T) – Тип дома. – House_Postal_Index (T) – Почтовый индекс дома. – Flat (T) – Номер квартиры. Правила: – Если найден один адрес, то метод вернёт параметры адреса. – Если не найдено адреса, то метод вернёт пустое значение. – Если найдено более одного адреса, то метод вернёт ошибку. – Если строка адреса не соответствует допустимым форматам, то метод вернёт ошибку. – В качестве сокращений части адреса допустимы любые из справочника dbo.KLSocr с учётом уровня/номера части адреса. – Сокращения могут быть в начале части адреса и в конце. – Сокращения могут отсутствовать. – Корпус дома берётся из строки адреса. – Номер квартиры берётся из строки адреса.
--	--

16 Работа с направлениями

16.1 Создание направления

POST api/EvnDirection

Метод создает в БД направление с переданными параметрами.

Правила метода:

1. Для параметра DirType_id:

- если DirType_id принимает одно из значений [10, 11, 15, 20], то параметры PrescriptionType_id, EvnPrescr_IsCito, обязателен;
- если DirType_id в значении 10, то PrescriptionType_id должен быть 11 или 12;
- если DirType_id в значении 11, то PrescriptionType_id должен быть 13;
- если DirType_id в значении 15, то PrescriptionType_id должен быть 6;
- если DirType_id в значении 20, то PrescriptionType_id должен быть 7;
- если DirType_id в значении 17, то:
 - параметры RequestConsult_id, ConsultationForm_id, ConsultingForm_id, RemoteConsultCause_id обязательные;
 - параметр EvnDirection_IsAgreement должен быть =2;
 - при создании направления:
 - в EvnDirection заполняются следующие поля:
 - Post_id – должность врача, обязательное для заполнения, если передан входящий параметр MedStaffFact_id, то вычисляется из места работы врача по входящему параметру MedStaffFact_id → v_MedsStaffFact.Post_id, иначе производится получение persis.Post.id по входящему параметру MedPost_Code=nsi.MedPost.MedPost_Code по соответствию nsi.MedPost.MedPost_id=persis.Post.MedPost_id;
 - EvnDirection_IsConcilium – если передан параметр EvnDirection_IsConcilium, то заполняется значением входящего параметра EvnDirection_IsConcilium, иначе не заполняется;
 - MedicalCareFormType_id – форма помощи, если:
 - ConsultationForm_id в значении 1, то указывается значение 3;
 - ConsultationForm_id в значении 2, то указывается значение 2;
 - ConsultationForm_id в значении 3, то указывается значение 1;
 - EvnDirection_IsConcilium – если передан параметр EvnDirection_IsConcilium, то заполняется значением параметра EvnDirection_IsConcilium, иначе не заполняется;
 - RecordType_id – источники записи, указывается значение 9 (иные источники записи);
 - EvnDirection_IsReceive – признак внешнего направления. Если передан параметр PassportToken_stid и не передан входящий параметр Lpu_sid, то заполняется значением 2;

- Org_sid – направившая организация. Если передан параметр PassportToken_std и не передан входящий параметр Lpu_sid, то заполняется значением fed.v_PassportToken.Org_id, где fed.v_PassportToken.PassportToken_tid равен входящему параметру PassportToken_std. Если организация не найдена, отображается ошибка с текстом: "Не найдена организация по параметру PassportToken_std", иначе заполняется значением v_Lpu.Org_id, где v_Lpu.Lpu_id равен входящему параметру Lpu_sid;
- Lpu_sid – направившая МО. Если передан параметр PassportToken_std и не передан входящий параметр Lpu_sid, то заполняется значением fed.v_PassportToken.Lpu_id, где fed.v_PassportToken.PassportToken_tid равен входящему параметру PassportToken_std, иначе заполняется входящим параметром Lpu_sid;
- MedService_id – служба ЦУК:
 - если направление на удаленную консультацию в формате "Врач-врач" (если DirType_id = 17 и RequestConsult_id=2 и EvnDirection_IsConcilium<>2), то из входящего параметра TimeTableGraf_id происходит получение действующей на текущую дату службы с типом "Центр удаленной консультации" на отделении врача. Если не найдена, то на группу отделений. Если не найдена на группе отделений, то на уровне МО. Иначе отображается ошибка с текстом: "Врач не устроен ни на одну службу с типом "Центр удаленной консультации";
 - если направление на удаленную консультацию в формате "Телемедицинский консилиум" (если DirType_id = 17 и EvnDirection_IsConcilium=2), то из входящего параметра TimetableMedService_id происходит получение данных. Иначе не заполняется;
- MedPersonal_did – если не передан входящий параметр MedPersonal_did, то происходит получение из бирки врача, к кому направили из входного параметра TimeTableGraf_id (TimeTableGraf.MedStaffFact_id=persis.WorkPlace.id → persis.WorkPlace.MedWorker_id), иначе заполняется входным параметром MedPersonal_did;
- MedPersonal_id – если передан входящий параметр MedStaffFact_id, то происходит получение из входного параметра MedStaffFact_id=persis.WorkPlace.id → persis.WorkPlace.MedWorker_id, иначе не заполняется;
- EvnDirection_confDT – если передан параметр TimetableMedService_id, то заполняется значением, иначе не заполняется;
- EvnDirection.Evn_id – создается событие выписки направления (EvnClass_id=27), где Evn.EvnStatus_id = 17 (Записано), статус направления;
- создается запись для DirType=17 со следующими полями:

- EvnDirection_id – идентификатор направления, обязательное для заполнения;
- TelemedEvnDirection_MedPersonalFIO – если передан параметр MedPersonal_FIO, то заполняется входящим параметром MedPersonal_FIO, иначе не заполняется;
- TelemedEvnDirection_MedPersonalExtId – если передан параметр MedPersonal_ExternalId, то заполняется входящим параметром MedPersonal_ExternalId, иначе не заполняется;
- если передан входящий параметр EvnDirection_IsConcilium=2, то создается запись о консилиуме (dbo.EvnConcilium):
 - EvnConcilium_id – идентификатор консилиума;
 - EvnConcilium_Num – порядковый номер консилиума, заполняется последним номером консилиума +1;
 - ConciliumCommunication_id – если входящий параметр ConsultingForm_id=1, то заполняется значением 1, иначе 2;
 - ConciliumForm_id – заполняется входящим параметром ConciliumForm_id;
 - ConciliumType_id – заполняется входящим параметром ConciliumType_id;
 - DiagSetPhase_id – заполняется входящим параметром DiagSetPhase_id;
 - EvnConcilium_IsTelemed – заполняется значением 2;
 - MedCarePlace_id – заполняется значением 8 (Дистанционно);
 - LpuSection_id – заполняется входящим параметром LpuSection_id;
 - EvnConsilium.Evn_id – создается событие Врачебный консилиум (EvnClass_id=220), где Evn.EvnStatus_id = 57 (Инициирован), статус консилиума, (связка с направлением через evn_pid);
- если DirType_id =10 и PrescriptionType_id =11, то параметры Evn_pid, UslugaComplexMedService_ResId обязательны.

2. Проверяются указанные ИД-параметры на наличие их в Системе:

- Evn_pid, Person_id, PayType_id, DirType_id, Diag_id, MedStaffFact_id, MedPersonal_zid, Lpu_sid, Lpu_did, LpuSection_did, LpuUnit_did, LpuSectionProfile_id, MedPersonal_did, PrescriptionType_id, MedService_id, Resource_id, StudyTarget_id, RemoteConsultCause_id, RequestConsult_id, ConsultingForm_id, ConsultationForm_id, LpuUnitType_id, UslugaComplex_id, TimetableGraf_id, TimetableStac_id, TimetableMedService_id, TimetableResource_id;
- если правило не выполняется, то метод возвращает соответствующую ошибку.

3. Параметр EvnLabRequest_id возвращается всегда, но имеет разные значения:

- если MedService_id не указан или служба НЕ имеет ник один из [lab, pzm, microbiolab], то выходной параметр EvnLabRequest_id будет =null;
- если MedService_id указан и служба имеет ни один из [lab, pzm, microbiolab], то создаётся заявка на исследование, связанная с созданным направлением, и возвращаемый EvnLabRequest_id будет не пустой.

4. Во входящих параметрах должен быть указан один из параметров TimeTableGraf_id, TimeTableStac_id, TimeTableMedService_id или TimeTableResource_id, иначе должен быть

указан параметр EvnQueue_id. Для этого необходимо вызвать метод добавления записи в лист ожидания POST api/EvnQueue/EvnQueue и в ответе получить обязательный параметр EvnQueue_id.

5. При создании направления предусмотрено сохранение RecMethodType_id = "РИШ".

Входящие параметры	– Evn_pid (N, Y) – идентификатор случая-родителя (движение в КВС, посещения в ТАП. Параметр обязательный для заполнения при DirType_id=10 и PrescriptionType_id=11; – Person_id (N, H) – идентификатор пациента. Регион: Москва: Параметр обязательный для заполнения при DirType_id = 7; – EvnDirection_Num (T, H) – номер направления; – EvnDirection_setDate (D, O) – дата направления; – PayType_id (N, O) – идентификатор вида оплаты; – DirType_id (N, O) – идентификатор типа направления; – Diag_id (N, O) – идентификатор диагноза; – EvnDirection_Descr (T, Y) – идентификатор обоснования направления. Параметр обязательный для заполнения, если DirType_id принимает одно из значений: – 1 "На госпитализацию плановую"; – 2 "На обследование"; – 3 "На консультацию"; – 4 "На восстановительное лечение"; – 5 "На госпитализацию экстренную"; – 6 "На осмотр с целью госпитализации"; – 10 "На исследование"; – 16 "На поликлинический прием"; – 17 "На удаленную консультацию"; – 31 "На осмотр перед вакцинацией". – Lpu_sid (N, Y) – идентификатор направившего МО. Параметр обязательный для заполнения: – для направлений на удаленную консультацию (если DirType_id = 17) в формате "Телемедицинский консилиум" (если DirType_id = 17 и EvnDirection_IsConcilium=2); – для направлений на удаленную консультацию в формате "Врач-врач" (если DirType_id = 17 и RequestConsult_id=2 и EvnDirection_IsConcilium<>2) и если не передан параметр PassportToken_stid; – для направлений если DirType_id <> 17; – PassportToken_stid (N, Y) – ОИД направившей организации. Обязательный параметр для заполнения для направлений на удаленную консультацию в формате "Врач-врач" (если DirType_id = 17 и RequestConsult_id=2 и EvnDirection_IsConcilium<>2) и если не передан параметр Lpu_sid; – LpuSection_id (N, H) – направившее отделение МО. Регион: Базовый (кроме Московской области); – MedStaffFact_id (N, Y) – идентификатор места работы направившего врача. Необязательный параметр для заполнения для направлений на удаленную консультацию в формате "Врач-врач" (если DirType_id = 17 и RequestConsult_id=2 и EvnDirection_IsConcilium<>2) или "Телемедицинский консилиум" (если DirType_id = 17 и EvnDirection_IsConcilium=2); – MedPersonal_zid (N, H) – идентификатор заведующего направившего
-------------------------------	---

	отделения; – Lpu_did (N, O) – идентификатор МО, куда направили; – LpuSection_did (N, H) – идентификатор отделения, куда направили; – LpuSectionProfile_id (N, O) – идентификатор профиля, куда направили. Для направлений на удаленную консультацию в формате "Врач-врач" (если DirType_id = 17 и RequestConsult_id=2 и EvnDirection_IsConcilium<>2) или "Телемедицинский консилиум" (если DirType_id = 17 и EvnDirection_IsConcilium=2) профиль отделения должен соответствовать профилю консультирования на службе ЦУК (dbo.LpuSectionProfileMedService.LpuSectionProfile_id по MedService_id). Если профиль отделения не соответствует, отображается сообщение: "Профиль отделения направления не соответствует профилю консультирования службы ЦУК"; – MedPersonal_did (N, H) – идентификатор врача, к кому направили; – LpuUnit_did (N, H) – идентификатор группы отделений, куда направили; – TimetableGraf_id (N, Y) – идентификатор поликлинической бирки. Обязательный параметр для заполнения для направлений на удаленную консультацию в формате "Врач-врач" (если DirType_id = 17 и RequestConsult_id=2 и EvnDirection_IsConcilium<>2). Если параметр не передан, отображается ошибка: "Отсутствуют обязательные параметры для направления на телемедицинскую консультацию: TimetableGraf_id"; – TimetableStac_id (N, H) – идентификатор стационарной бирки; – TimetableMedService_id (N, Y) – идентификатор бирки на службу. Обязательный параметр для заполнения для направлений на удаленную консультацию в формате "Телемедицинский консилиум" (если DirType_id = 17 и EvnDirection_IsConcilium=2). Если параметр не передан, то отобразится ошибка: "Отсутствуют обязательные параметры для направления на телемедицинскую консультацию: TimetableMedService_id"; – TimeTableResource_id (N, H) – идентификатор диагностической бирки; – EvnPrescr_id (N, H) – идентификатор назначения. Регион: Базовый (кроме Московской области); – PrescriptionType_id (N, Y) – идентификатор типа назначения. Параметр обязательный для заполнения, если DirType_id принимает одно из значений [10, 11, 15, 20]; – EvnPrescr_IsCito (boolean, Y) – признак "Cito" (0 – Нет, 1 – Да). Обязательный параметр, если DirType_id = 10, 11, 15 или 20; – MedService_id (N, H) – идентификатор службы; – MedService_lid (N, H) – идентификатор пункта забора крови. Передача параметра должна быть связана с UslugaComplexMedService_ResId через параметр MedServiceLinkType_id = 1. Если условие не выполняется, отображается ошибка: "Связанная служба не является пунктом забора". Учитывается, если DirType_id=10, в другом случае игнорируется; – Resource_id (N, H) – идентификатор ресурса; – fromMobile (Boolean, H) – признак мобильного устройства; – StudyTarget_id (N, H) – цель исследования; – RemoteConsultCause_id (N, Y) – цель удаленной консультации.
--	---

	Параметр обязательный, если DirType_id = 17; – RequestConsult_id (N, Y) – идентификатор пациента, которому требуется консультация. Параметр обязательный, если DirType_id = 17; – ConsultingForm_id (N, Y) – идентификатор режима оказания консультации. Параметр обязательный, если DirType_id = 17; – ConsultationForm_id (N, Y) – идентификатор формы оказания консультации. Параметр обязательный, если DirType_id = 17; – EvnDirection_IsConcilium (N, Y) – признак "Требуется консилиум". Параметр обязательный для заполнения для направлений на удаленную консультацию в формате "Врач-врач" (если DirType_id = 17 и RequestConsult_id=2). Допустимые значения: 1 – Нет, 2 – Да. Если введено недопустимое значение, то отобразится сообщение: "Введено неверное значение признака "Требуется консилиум" EvnDirection_IsConcilium"; – PersonPhone_Phone (T, H) – номер телефона; – EvnDirection_IsAgreement (N, Y) – идентификатор добровольного информированного согласия на ТМК. Обязательный параметр, если DirType_id = 17 и значение =2. Необходимо передавать EvnDirection_IsAgreement='2', в другом случае отображается ошибка: "Отсутствуют обязательные параметры для направления на телемедицинскую консультацию: EvnDirection_IsAgreement"; – LpuUnitType_id (N, H) – идентификатор условия оказания медицинской помощи. Для региона Базовый (кроме Московской области); – UslugaComplex_id (N, H) – идентификатор услуги; – UslugaComplexMedService_ResId (N, Y) – идентификатор исследования. Обязателен для заполнения при создании направления на лабораторное исследование (DirType_id = 10 И PrescriptionType_id = 11). Определяется через UslugaComplexMedService_pid (UslugaComplex_id). Для региона Базовый (кроме Московской области); – RecMethodType_id (N, Y) – идентификатор способа записи. Обязательный параметр для заполнения для направлений на удаленную консультацию в формате "Врач-врач" (если DirType_id = 17 и RequestConsult_id=2 и EvnDirection_IsConcilium<2) или "Телемедицинский консилиум" (если DirType_id = 17 и EvnDirection_IsConcilium=2). Если направление на удаленную консультацию (DirType_id = 17), то допустимы следующие значения: 20 (Доктис), 13 (РИШ). Иначе допустимо только значение 13 (РИШ). Если введено недопустимое значение, то должно отобразится сообщение: "Введено неверное значение способа записи RecMethodType_id". Если параметр не передан, то по умолчанию устанавливается значение 13 (РИШ); – MedPersonal_FIO (T, Y) – Ф. И. О. направившего врача . Обязательный параметр для заполнения, если не передан входящий параметр MedStaffFact_id для направлений на удаленную консультацию в формате "Врач-врач" (если DirType_id = 17 и RequestConsult_id=2 и EvnDirection_IsConcilium<2) или "Телемедицинский консилиум" (если DirType_id = 17 и EvnDirection_IsConcilium=2)"; – MedPost_Code (T, Y) – код должности направившего врача из справочника "Должности работников организаций медицинского и фармацевтического профиля 1.2.643.5.1.13.13.11.1002". Обязательный
--	---

	параметр для заполнения, если: – не передан входящий параметр MedStaffFact_id; – для направлений на удаленную консультацию в формате "Врач-врач" (если DirType_id = 17 и RequestConsult_id=2 и EvnDirection_IsConcilium<>2) или "Телемедицинский консилиум" (если DirType_id = 17 и EvnDirection_IsConcilium=2); – MedPersonal_ExternalId (Т, Н) – идентификатор врача из внешней системы; – ConciliumType_id (N, У) – идентификатор профиля консилиума. Обязательное для заполнения для направлений на удаленную консультацию в формате "Телемедицинский консилиум" (если DirType_id = 17 и EvnDirection_IsConcilium=2). Допустимы значения из справочника dbo.ConciliumType. Если введено недопустимое значение, то должно отобразиться сообщение: "Введено неверное значение профиля консилиума ConciliumType_id"; – CommissionTarget_id (N, У) – идентификатор цели консилиума. Обязательное для заполнения для направлений на удаленную консультацию в формате "Телемедицинский консилиум" (если DirType_id = 17 и EvnDirection_IsConcilium=2). Допустимы значения из справочника nsi.CommissionTarget с признаком CommissionTarget_IsTelemedConcilium=2. Если введено недопустимое значение, то должно отобразиться сообщение: "Введено неверное значение цели консилиума CommissionTarget_id"; – ConciliumForm_id (N, У) – идентификатор вида проведения консилиума. Обязательное для заполнения для направлений на удаленную консультацию в формате "Телемедицинский консилиум" (если DirType_id = 17 и EvnDirection_IsConcilium=2). Допустимы значения из справочника nsi.ConciliumForm. Если введено недопустимое значение, то должно отобразиться сообщение: "Введено неверное значение вида проведения консилиума ConciliumForm_id"; – DiagSetPhase_id (N, У) – идентификатор общего состояния пациента – TestList(A, Н) – список тестов исследования; если массив не заполнен, то в направлении должны быть назначены все тесты исследования; – UslugaComplexMedService_id (N, О) – идентификатор теста. – EvnDirectionOper_IsAgree (N, Н) – идентификатор согласия на операцию. Параметры для направления на прижизненное патологоанатомическое исследование биопсийного (операционного) материала. Данные поля должны быть заполнены в случае, если DirType_id = 7. Регион: Московская область – LpuSectionBedProfile_id (N, Н) – идентификатор профиля коек; – EvnDirectionHistologic_IsUrgent (Boolean, Н) – срочность; – Lpu_aid (N, Н) – в патологоанатомическую лабораторию МО; – LpuSectionName (Т, Н) – наименование отделения; – MedPersonalFIO (Т, Н) – Ф. И. О. направившего врача; – MedPersonalPhoneNum (Т, Н) – номер телефона врача; – PersonAmbulatCard_id – (N, Н) – идентификатор оригинала
--	--

	амбулаторной карты; – PersonAmbulatCard_Num (N, H) – номер оригинала амбулаторной карты; – EvnPS_id (N, H) – идентификатор КВС; – PredOperTreat (T, H) – проведенное предоперационное лечение; – ClinicalDiag(T, H) – клинический диагноз; – ClinicalData (T, H) – клинические данные; – HistologicMaterial_id (N, H) – вид материала; – BiopsyOrder_id (N, H) – кратность биопсии; – BiopsyDT (DT, H) – дата первичной биопсии; – BiopsyNum (T, H) – номер первичной биопсии; – Operation (T, H) – вид операции; – SpecimenSaint (T, H) – маркировка материала; – ObjectCount (N, H) – число объектов; – BiopsyReceive_id (N, H) – способ получения материала; – IsPlaceSolFormalin (Boolean, H) – материал помещен в 10% раствор нейтрального формалина; – Evn_didDT (DT, O) – дата и время операции (забора материала); – LpuSection_id (N, O) – идентификатор направившего отделения МО; – MedPersonal_id (N, O) – идентификатор направившего врача
Выходные параметры	Если запрос обработан успешно, и в БД создана запись, то в ответе будет отправлено: – EvnDirection_id (N, H) – идентификатор направления на лабораторное исследование; – EvnConcilium_id (N, H) – идентификатор консилиума; – Evn_id (N, H) – идентификатор события; – EvnQueue_id (N, H) – идентификатор постановки в очередь; – EvnLabRequest_id (N, H) – идентификатор заявки на лабораторное исследование; – EvnPrescr_id (N, H) – идентификатор назначения. Если входящий параметр DirType_id = 10, 11, 15 или 20, то создается назначение, связанное с направлением и таким же случаем-родителем Evn_pid, в ответе обязательный параметр EvnPrescr_id; – EvnDirectionHistologic_id (N, H) – идентификатор в таблице направления на прижизненное патологоанатомическое исследование биопсийного (операционного) материала. Регион: Москва: Если входящий параметр DirType_id = 7, то создается запись в таблице, в ответе – обязательный параметр EvnDirectionHistologic_id
Пример	<pre>{ "error_code": 0, "data": { "EvnDirection_id": "140101014227728", "Evn_id": "140101014227728", "EvnLabRequest_id": "140101014227729", "EvnQueue_id": "140101014227727", "EvnPrescr_id": "140101014227726" } }</pre>

17 Лист ожидания

17.1 Добавление записи в лист ожидания

POST api/EvnQueue/EvnQueue

Входящие параметры	- Person_id (N, O) – идентификатор пациента; - LpuSectionProfile_id (N, Y) – идентификатор профиля отделения МО. - LpuSection_id (N, H) – идентификатор отделения. - MedPersonal_id (N, H) – направивший врач; - MedStaffFact_did (N, Y) – врач, к кому в лист ожидания включается пациент. Указывается, если запись выполняется к конкретному врачу. - Lpu_sid (N, H) – направившее МО; - Lpu_did (N, O) – МО, куда направили; - EvnQueue_desDT (D, H) – дата и время посещения (желаемая); - MedService_did (N, Y) – идентификатор службы, в которую нужно поставить в очередь; - UslugaComplex_did (N, O) – идентификатор услуги, на которую необходимо записать
Действия при получении запроса	а) Выполняется проверка заполнения обязательных полей, в т.ч.: должен быть заполнен отделение или профиль, или врач, к которому в очередь ставится пациент. Если ни профиль, ни врач не указаны, то ошибка. б) Если по переданному значению Person_id не найден пациент, то ошибка "Пациент не найден в системе". в) Если запись с аналогичными параметрами найдена в системе, то ошибка "Данные записи не прошли проверку на дублирование". Создание записи не производится. г) Если в любом передаваемом параметре передается значение, не соответствующее ожидаемому формату, например: - ожидается числовое значение, а передано текстовое значение; - ожидается параметр, состоящий из 11 символов, а передан параметр, состоящий из большего или меньшего количества символов; то, ошибка: "Неверный формат поля <Наименования полей через запятую>", создание записи в БД не происходит. д) Если в передаваемом параметре (LpuSectionProfile_id, LpuSection_id, MedStaffFact_did, Lpu_sid, Lpu_did) передается значение id, не найденное в системе, ошибка: "Значения для <Наименования полей через запятую> не найдены в справочниках системы". Создание записи в БД не происходит. Если есть ошибки, то возвращается ошибка. Иначе: - в лист ожидания добавляется запись – пациент ставится в очередь по профилю или врачу (в соответствии с полученными параметрами). - статус записи "в очереди" (EvnQueueStatus_id). - способ записи/включения в Лист ожидания – "РИШ"
Ответ	Если есть ошибки, то возвращается код ошибки. Иначе: - EvnQueue_id (N, O) – идентификатор постановки в очередь - EvnQueueStatus_id (N, H) – статус листа ожидания

17.2 Изменение статуса записи в листе ожидания

PUT api/EvnQueue/EvnQueueStatus

Входящие параметры	– EvnQueue_id (N, O) – идентификатор постановки в очередь – EvnQueueStatus_id (N, O) – статус записи в очереди (может принимать значение Отмена). – QueueFailCause_id (N, O) – идентификатор причины изменения порядка в очереди. – EvnStatus_id (N,O) – идентификатор статуса направления. Может принимать значения 12 (отменено) или 13 (отклонено). – MedPersonal_id (N, O) – врач, изменивший статус записи в листе ожидания (значение сущности persis.MedWorker);
Действия	а) Выполняется проверка заполнения обязательных параметров, в т.ч. – если не заполнен ни один из условных параметров или обязательный параметр, то ошибка. б) Если по переданному значению EvnQueue_id не найдена запись о постановке в очередь, то ошибка "Запись о постановке в очередь с указанным EvnQueue_id не найдена в системе". в) Если в любом передаваемом параметре передается значение, не соответствующее ожидаемому формату, например: – ожидается числовое значение, а передано текстовое значение; – ожидается параметр, состоящий из 11 символов, а передан параметр, состоящий из большего или меньшего количества символов; то, ошибка: "Неверный формат поля <Наименования полей через запятую>", обновление статуса записи в БД не происходит. г) Если в передаваемом параметре (EvQueueStatus_id, QueueFailCause_id, EvnStatus_id, MedPersonal_id) передается значение id, не найденное в системе, ошибка: "Значения для <Наименования полей через запятую> не найдены в справочниках системы". Обновление статуса записи в БД не происходит. Если ошибок нет, то выполняется изменение данных листа ожидания. Если полученные параметры статус листа ожидания, статус направления отличаются от вышеуказанных возможных значений, то возвращается ошибка. Иначе: Лист ожидания отменяется. Если листу ожидания уже предложена запись на прием, то запись отменяется.
Ответ	Успешный ответ – пустой ответ с кодом ошибки "0"

17.3 Изменение листа ожидания

PUT api/EvnQueue/EvnQueue

Изменение листа ожидания. Метод используется для отмены листа ожидания.

Входящие параметры	– EvnQueue_id (N, O) – идентификатор постановки в очередь – EvnQueueStatus_id (N, O) – статус листа ожидания
Действия	Выполняется проверка заполнения обязательных параметров, в т.ч. – если не заполнен ни один из условных параметров или обязательный параметр,

	то ошибка. Если ошибок нет, то выполняется изменение данных листа ожидания. Если полученных параметр статус листа ожидания не равен "Отмена", то возвращается ошибка. Иначе: Лист ожидания отменяется. Если листу ожидания уже предложена запись на прием, то запись отменяется, причина отмены – по инициативе пациента.
Ответ	Если есть ошибки, то возвращается код ошибки и ошибка. Иначе: выдаются следующие данные: – EvnQueue_id (N, O) – идентификатор постановки в очередь Person_id (N) – Идентификатор пациента. – Lpu_id (N) – идентификатор МО очереди в справочнике РИШ. – LpuSectionProfile_id (N) – идентификатор профиля отделения МО. – LpuSection_id (N) – идентификатор отделения – MedStaffFact_id (N) – врач, к кому в лист ожидания включен пациент (значение сущности persis.MedWorker); – EvnQueue_desDT (D, H) – желаемые дата и время посещения. – EvnQueue_setDate (D) – дата постановки в очередь; – EvnQueue_setTime (T) – время постановки в очередь – EvnQueueStatus_id (N) – статус записи в очереди. – EvnQueue_failDT (D) – дата отмены листа ожидания

17.4 Получение листа ожидания

GET api/EvnQueue/EvnQueue

Входящие параметры	– Person_id (N, Y) – идентификатор пациента. Обязательно указывается, если не заполнены другие условные параметры; – Lpu_id (N, Y) – идентификатор МО в справочнике РИШ. Обязательно указывается, если не указан пациент; – LpuSection_id (N, Y) – идентификатор отделения. – LpuSectionProfile_id (N, Y) – идентификатор профиля отделения МО. Обязательно указывается, если не указан пациент и отделение и врач. – MedStaffFact_id (N, Y) – врач, к кому в лист ожидания включается пациент.)
Действия	Выполняется проверка заполнения обязательных параметров, в т.ч. – если не заполнен ни один из условных параметров или обязательный параметр, то ошибка. Если есть ошибки то, возвращается код ошибки. Иначе: Возвращаются записи очереди, удовлетворяющие заданным входящим параметрам – имеют статус "в очереди"; – или у которых направления имеют статус "Поставлено в очередь".
Ответ	Если есть ошибки, то возвращается код ошибки. Иначе: выдаются следующие данные: – EvnQueue_id (N, O) – идентификатор постановки в очередь Person_id (N) – Идентификатор пациента. – Lpu_id (N) – идентификатор МО очереди в справочнике РИШ. – LpuSectionProfile_id (N) – идентификатор профиля отделения МО. – LpuSection_id (N) – идентификатор отделения – MedStaffFact_id (N) – врач, к кому в лист ожидания включен пациент (значение сущности persis.MedWorker);

	– EvnQueue_setDate (D) – дата постановки в очередь; – EvnQueue_setTime (T) – время постановки в очередь – EvnQueueStatus_id (N) – статус записи в очереди. – EvnQueue_desDT (D, H) – желаемые дата и время посещения
--	--

18 Общие методы

18.1 Поиск человека по атрибутам

GET /api/Person/mgetPersonSearch

Мобильный метод предназначен для поиска человека по атрибутам

Входящие параметры	Условия формирования параметров запроса: – Если передается единственный параметр - номер КВС, то он обязательный; – Если передается Идентификатор МО, то обязательно передавать еще номер КВС; – Если Идентификатор МО не передается, то номер КВС не обязательно передавать; – Если передается другой параметр запроса (не Идентификатор МО и не номер КВС), тогда надо обязательно указать два параметра запроса; – Возможные значения в параметре запроса searchMode (по умолчанию=all): – wow - цепляем участников ВОВ; – attachrecipients - только по льготникам и прикрепленным; – evnpldispscreen - если ищем для скринингового исследования, то возраст пациента на конец выбранного года должен соответствовать – evnpldtipre - если ищем по картам первого этапа предвостмотра несовершеннолетних – evnpldtipro - если ищем по картам первого этапа профосмотра несовершеннолетних – withlgotonly - только по льготникам – dddispclass2 - ДВН 2 – ddorpperiod - периодический осмотр – ddorpsec - если ищем по картам первого этапа детей-сирот – hasopenevnps - если ищем по открытым КВС – ddorp или ddorp_all - с и ищем по регистру детей-сирот – geriatrics - старше 60 лет – dt14 - если ищем по ДД 14 – older14notdead - только старше 14 лет и не умершие – dt6 - дети младше 6 лет – palliat или narc - хз, но вроде ищем без даты смерти – attbefore3 - только прикрепленные или старше 3 лет – att_vol - только прикрепленные и в объеме – att_1_4 - только прикрепленные (основное или служебное) – att - только прикрепленные – person_register_by_vzn_request - поиск в регистре для заявки ВЗН – hms - поиск для регистра главных внештатных специалистов – fmba - только не включенные в регистр ФМБА – dd - если ищем по ДД – encryonly - отображение только зашифрованных ВИЧ-инфицированных – men_only - women_only - только эм или жо – smo или smo3 - Поиск в арм смо. Для Перми - последние 3 года – erssnils - снилс должен быть заполнен
---------------------------	---

Пример	Успешный ответ: <pre> { "error_code": 0, "data": [{ "Person_id": "100126424", "Server_id": "3", "PersonEvn_id": "88401919", "Person_IsInErz": null, "Person_Phone": null, "Person_Inn": null, "Person_IsUnknownSign": null, "Person_isOftenCaller": 1, "Person_IsRefuse": "true", "Person_IsDead": "false", "Person_IsFedLgot": "false", "Person_IsRegLgot": "false", "Person_Lgots": "false", "Person_Is7Noz": "false", "UAddress_AddressText": null, "PAddress_AddressText": null, "Person_IsBDZ": null, "PersonQuarantine_begDT": null, "PersonQuarantine_IsOn": "false", "PersonCard_IsDms": "false", "PersonSurName_SurName": "ИВАНОВ", "PersonFirName_FirName": "АЛЕКСАНДР", "PersonSecName_SecName": "АЛЕКСЕЕВИЧ", "Polis_Ser": null, "PolisFormType_id": null, "OrgSMO_id": null, "OMSSprTerr_id": null, "Polis_endDate": null, "Person_Snils": "06252245839", "Document_begDate": null, </pre>

	<pre> "OrgDep_Text": null, "Polis_Num": null, "Document_Ser": null, "Document_Num": null, "Polis_EdNum": null, "Person_Age": 70, "PersonBirthDay_BirthDay": "01.06.1951", "Person_deadDT": null, "Sex_id": "1", "Lpu_Nick": null, "CmpLpu_id": null, "PersonCard_Code": null, "accessType": "edit" }, { "Person_id": "4006173", "Server_id": "1", "PersonEvn_id": "57406108", "Person_IsInErz": null, "Person_Phone": null, "Person_Inn": null, "Person_IsUnknownSign": null, "Person_isOftenCaller": 1, "Person_IsRefuse": "false", "Person_IsDead": "false", "Person_IsFedLgot": "false", "Person_IsRegLgot": "false", "Person_Lgots": "false", "Person_Is7Noz": "false", "UAddress_AddressText": null, "PAddress_AddressText": null, "Person_IsBDZ": null, "PersonQuarantine_begDT": null, "PersonQuarantine_IsOn": "false", "PersonCard_IsDms": "false", "PersonSurName_SurName": "ИВАНОВ", "PersonFirName_FirName": "ВИТАЛИЙ", </pre>
--	--

	<pre> "PersonSecName_SecName": "ПОЛИКАРПОВИЧ", "Polis_Ser": null, "PolisFormType_id": null, "OrgSMO_id": null, "OMSSprTerr_id": null, "Polis_endDate": null, "Person_Snils": "02756392269", "Document_begDate": null, "OrgDep_Text": null, "Polis_Num": null, "Document_Ser": null, "Document_Num": null, "Polis_EdNum": null, "Person_Age": 70, "PersonBirthDay_BirthDay": "01.06.1951", "Person_deadDT": null, "Sex_id": "1", "Lpu_Nick": null, "CmpLpu_id": null, "PersonCard_Code": null, "accessType": "edit" }] } </pre>
Ответ	Логика формирования параметров ответа: Person_IsBDZ. Значение по умолчанию: – Если Polis_endDate не пустое поле и Polis_endDate меньше текущей даты, то 'orange'. – Если person.PersonCloseCause_id= 2 и person.Person_closeDT не пустое поле, то 'red' – Если person.server_id = 0, то 'true' Иначе 'false' – Для Перми: – Если person.server_id = 0, то 'true' – Если person.Person_IsInErz = 1, то 'blue' – Если person.server_id = 0 и Polis_endDate меньше текущей даты, то – Если Person_deadDT - пустое поле, то 'yellow' - Иначе - 'red' – Если person.server_id = 2 и person.Person_IsInErz не равно 1, то 'false'

	– Для Пензы: – Если person.Person_IsInErz = 1, то 'orange' – Если person.Person_IsInErz = 2, то 'true' – Иначе 'false' – Для Казахстана - – Если person.Person_IsInErz = 1, то 'red' – Если person.Person_IsInErz = 2, то 'true'Иначе 'false'
Ошибки	Если не указаны параметры запроса: <pre>{ "success": false, "error_msg": "Не указан ни один фильтр. Для поиска заполните хотя бы один параметр в блоке фильтров.", "error_code": 6 }</pre>

19 Вызов СМП SWAN API

19.1 Метод добавления талона вызова

POST /emergency/EmergencyCallCard

Метод предназначен для добавления талона вызова

Входящие параметры	– CmpCallCard – талон вызова: – Guid (T); – primaryCallCard.id (long) – ссылка на первичный вызов; – number (long, Y) – номер вызова за день; – numberYear (long, Y) – номер вызова за год; – priority (N) – приоритет; – region.id (long) – регион адреса вызова (KLArea); – subRegion.id (long) – район адреса вызова (KLArea); – city.id (long) – город адреса вызова (KLArea); – town.id (long) – населенный пункт адреса вызова (KLArea); – street.id (long) – улица адреса вызова (KLStreet); – streetSecond.id (long) – 2-я улица перекрестка (KLStreet); – house (T (30)) – номер дома адреса; – corpus (T (5)) – номер корпуса адреса; – flat (T (5)) – номер квартиры; – room (T (5)) – номер комнаты; – entrance (T (200)) – номер подъезда. Размерность параметра 200 символов; – floor (T (200)) – номер этажа. Размерность параметра 200 символов; – entranceCode (T (200)) – код домофона. Размерность параметра 200 символов; – phone (T (20)) – номер телефона; – comment (T (4000)) – комментарий; – reason.id (long) – повод к вызову (CmpReason). Если в регионе подача счетов на оплату через реестры осуществляется через Систему, то используется признак необходимости включения карты вызова в реестры на оплату; – reasonAdditionalInfo (T (1000)) – текстовое описание повода вызова. Расширенная информация с текстовым значением повода к вызову (если указан соответствующий reason.id), либо текстовое описание повода к вызову и доп. информации к нему – person.id (long, O) – идентификатор человека. Person_id (для неидентифицированных пациентов создается неизвестный человек); – surname (T (30)) – фамилия пациента; – firname (T (30)) – имя; – secname (T (30)) – отчество; – age (N) – возраст; – ageDays (N) – возраст (дней);
----------------------------------	--

	– ageMonths (N) – возраст (полных месяцев); – birthday (S (DT) (23)) – дата рождения; – sex.id (N) – идентификатор половой принадлежности (Sex); – polis.series (T (10)) – серия полиса ОМС; – polis.number (T (30)) – номер полиса ОМС; – polis.federalNumber (T (16)) – единый номер полиса ОМС; – whoCallType.id (long) – вид вызывающего помощь; – whoCall (T (30)) – кто вызывает помощь; – callType.id (long) – тип вызова (CmpCallType); – callProfile.id (long) – профиль вызова (CmpProfile); – urgency (N) – срочность вызова; – acceptTime (T (DT) (23), Y) – дата и время получения вызова; – endingTime (T (DT) (23), Y) – дата и время окончания вызова; – arrivalPlanTime (T (DT) (23)) – плановое время прибытия на место вызова; – arrivalTime (T (DT) (23)) – время прибытия на место вызова; – hospitalArrivalTime (T (DT) (23)) – время прибытия в стационар; – transportTime (string (datetime) (23)) – время начала транспортировки; – departureTime (string (datetime) (23)) – время выезда на вызов; – acceptLine (integer) – пульт приема; – callDuration (integer) – длительность приема вызова (сек); – emergencyTeam.id (long, Y) – идентификатор бригады вызова (EmergencyTeam); – emergencyTeamSpec.id (long) – идентификатор профиля бригады СМП; – emergencyTeamNum (string (32)) – номер бригады; – medstaffTeam.id (long, Y) – врач бригады СМП; – transferTime (string (datetime) (23)) – время передачи вызова; – lpu.id (long, Y) – МО обслуживания вызова (Lpu); – lpuAccept.id (long) – МО принятия вызова (Lpu); – lpuTransferUrgent.id (long) – МО передачи вызова (НМП); – lpuTransferEmergency.id (long) – МО передачи вызова (СМП); – lpuBuilding.id (long) – подразделение СМП (LpuBuilding); – lat (number) – широта адреса вызова; – lon (number) – долгота адреса вызова; – diag.id (long, Y) – основной диагноз; – diag.code (string (10), Y) – код диагноза по МКБ; – isUrgent (boolean) – признак неотложной помощи; – isOpen (Boolean) – признак открытой карты; – isActiveToPolyclinic (Boolean) – признак актива в поликлинику; – callPlaceType.id (long) – тип места вызова
--	---

	(CmpCallPlaceType); – nsiCmpCallPlaceType.id (long) – место оказания СМП; – lpuHospitalization.id (long, Y) – МО госпитализации (Lpu); – address.fiasguid (string (uniqueidentifier)) – ГУИД кода ФИАС адреса места вызова – address.fiascode (string (33),Y) – код ФИАС адреса места вызова; – addressTxt (string(1000), H) – текстовое описание адреса вызова; – commonState.id (long) – идентификатор степени тяжести состояния; – callSignType.code (integer) – тип вызова СМП; – stationNum (integer) – подстанция СМП (номер); – transportation.features (String) – особенности, связанные с транспортировкой и оказанием скорой медицинской помощи больному; – numberCS (string (255)) – номер сопроводительного листа; – document.number (string(30), H) – серия и номер документа удостоверяющего личность; – document.orgDepText (string(1000), H) – организация, выдавшая документ удостоверяющий личность; – document.begDate (datetime, H) – дата выдачи документа удостоверяющего личность; – uaddress (string (1000), H) – адрес места жительства пациента; – accidentCircumstance (string(512), H) – обстоятельства несчастного случая; – CmpEmTeamDuty: – emergencyTeamWatch.id (long) – дежурство бригады СМП; – isGenerateNumber (Boolean) – признак генерации номера вызова. Всегда должен быть True. Если методом передано значение параметра отличное от True, то значение игнорируется и считается = True, если в настройках конфигурации API значение параметра isIntegration = " true" (см. swan-api)
Ответ	Пример ответа: <pre>{ "errorCode": 0, "errorMessage": "OK", "rowCount": 1, "responseData": { "errorCode": 0, "errorMessage": "OK",</pre>

	<pre>"id": 590910000083170, "guid": null, "stringId": null, "ids": null, "version": null, "versionOld": null, "complex": false, "success": true }, "success": true }</pre>
--	--

20 Аварийные ситуации

20.1 Описание аварийных ситуаций

Надежность Системы обеспечивается при следующих аварийных ситуациях:

- отказ Системы;
- сбой Системы.

Отказом Системы следует считать событие, состоящее в утрате работоспособности Системы и приводящее к невыполнению или неправильному выполнению контрольных примеров или задач функциональных модулей.

Сбоем Системы следует считать событие, состоящее во временной утрате работоспособности Системы и характеризующее возникновением ошибки при выполнении контрольных примеров или задач функциональных модулей.

В Системе предусмотрено автоматическое восстановление обрабатываемой информации в следующих аварийных ситуациях:

- программный сбой при операциях записи–чтения;
- разрыв связи с клиентской программой (терминальным устройством) в ходе редактирования/обновления информации.

В Системе предусмотрена возможность ручного восстановления обрабатываемой информации из резервной копии в следующих аварийных ситуациях:

- физический выход из строя дисковых накопителей;
- ошибочные действия обслуживающего персонала.

В Системе предусмотрено автоматическое восстановление работоспособности серверной части Системы в следующих ситуациях:

- штатное и аварийное отключение электропитания серверной части;
- штатная перезагрузка Системы и загрузка после отключения;
- программный сбой общесистемного программного обеспечения, приведший к перезагрузке Системы.

В Системе предусмотрено полуавтоматическое восстановление работоспособности серверной части Системы в следующих аварийных ситуациях:

- физический выход из строя любого аппаратного компонента, кроме дисковых накопителей – после замены компонента и восстановления конфигурации общесистемного программного обеспечения;
- аварийная перезагрузка системы, приведшая к нефатальному нарушению целостности файловой системы – после восстановления файловой системы.

Для восстановления Системы после отказа или сбоя, необходимо сначала устранить причину отказа/сбоя (заменить неисправное оборудование, устранить системные ошибки и др.), а затем предпринять следующие действия:

- установить операционную систему, а затем – соответствующий пакет обновления; проверить правильность работы домена.
- установить СУБД, а затем – соответствующий пакет обновления.
- восстановить базу данных из резервной копии; перезагрузить сервер после восстановления базы данных.
- проверить доступность Системы; чтобы убедиться в правильности работы, запустите сценарий проверки основных функций.
- активировать возможность работы пользователей в штатном режиме.

В случае отказа или сбоя Системы, связанного с неисправностью оборудования, работы проводит Администратор Заказчика.

В случае отказа или сбоя Системы, связанного с системной ошибкой, работы проводит Администратор Исполнителя.

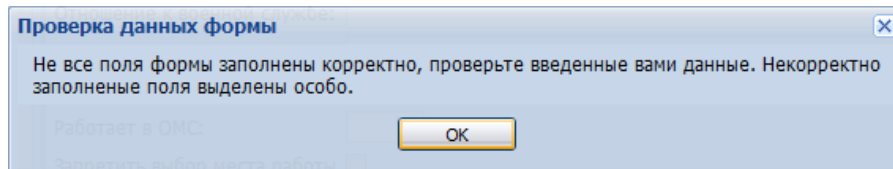
20.2 Действия в случае несоблюдения условий выполнения технологического процесса

При работе с Системой пользователю могут отображаться сообщения нескольких типов:

- сообщение об успешном завершении действия;
- сообщение об ошибке;
- предупреждение;
- сообщение о неисправности системы.

Сообщение об успешном завершении действия содержит краткое резюме операции. Для закрытия сообщения нажмите кнопку **"ОК"**.

Сообщение об ошибке отображается в случае, когда дальнейшее выполнение действия в Системе невозможно. Как правило, в таком сообщении содержится краткое описание причины возникновения ошибки. Для закрытия сообщения об ошибке нажмите кнопку **"ОК"**.



Предупреждение отображается в том случае, если действия, совершенные оператором, могут повлечь за собой какие-либо особенности в выполнении операции, но не приведут к ошибке. Например, если оператор укажет у сотрудника ставку менее 0,1, то отобразится сообщение, что такая ставка не будет учитываться при выгрузке. Для того чтобы продолжить выполнение действия, нажмите кнопку **"Да"/"Продолжить"**. Для того чтобы прекратить действие, нажмите кнопку **"Нет"/"Отмена"**.

В случае возникновения ошибки о неисправности системы, пользователю системы следует обратиться к администратору системы.

Администратор системы для решения проблем обращается к эксплуатационной документации, настоящему руководству, онлайн справочной системе.

В случае невозможности разрешения ситуации следует обратиться в техническую поддержку.