

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_1

Сервис "Вызов врача на дом" 3.0.3_1

Содержание

| | | |
|-------|--|----|
| 1 | Введение | 3 |
| 1.1 | Область применения | 3 |
| 1.2 | Краткое описание возможностей | 3 |
| 1.3 | Уровень подготовки администратора..... | 3 |
| 1.4 | Перечень эксплуатационной документации, с которым необходимо ознакомиться администратору | 3 |
| 2 | Общие сведения | 4 |
| 2.1 | Описание информационного взаимодействия..... | 4 |
| 2.2 | Схема взаимодействия | 4 |
| 2.3 | Кэширование запросов..... | 4 |
| 2.4 | HTTP ошибки..... | 4 |
| 2.4.1 | Ошибки на стороне клиента..... | 5 |
| 2.4.2 | Ошибки на стороне сервера | 5 |
| 2.4.3 | Детализация ошибок | 5 |
| 3 | Авторизация | 7 |
| 3.1 | Авторизация в PHP API. Авторизация в системе..... | 7 |
| 3.2 | Авторизация в SWAN API. Метод авторизации пользователя СМП..... | 8 |
| 4 | Описание сервиса "Вызов врача на дом" | 9 |
| 5 | Описание методов..... | 11 |
| 5.2 | Общие параметры для всех методов | 11 |
| 5.3 | Описание параметров метода..... | 11 |
| 5.4 | Описание общих кодов ошибок | 12 |
| 5.5 | Описание методов работы с человеком | 12 |
| 5.5.1 | Создание человека POST api/Person..... | 12 |
| 5.5.2 | Редактирование данных человека PUT api/Person | 17 |
| 5.5.3 | Поиск человека по атрибутам GET /api/Person/mgetPersonSearch | 21 |
| 5.6 | Методы сервиса "Вызов врача на дом"..... | 26 |
| 5.6.2 | Добавление вызова врача на дом POST api/HomeVisit/HomeVisit | 26 |
| 5.6.3 | Получение адреса GET api/Address | 35 |
| 5.6.4 | Получение списка параметров адреса GET api/AddressSearch | 38 |
| 5.6.5 | Метод добавления талона вызова POST /emergency/EmergencyCallCard..... | 40 |
| 6 | Аварийные ситуации | 45 |
| 6.1 | Описание аварийных ситуаций..... | 45 |
| 6.2 | Действия в случае несоблюдения условий выполнения технологического процесса .. | 46 |

1 Введение

1.1 Область применения

Настоящий документ описывает порядок работы с сервисом "Вызов врача на дом" 3.0.3_1 подсистемы "Взаимодействие с внешними голосовыми ботами (в т.ч. технологии ИИ) 122" 3.0.3_1 (далее – Подсистема, подсистема), являющейся частью Единой информационной системы здравоохранения (далее–Система, система).

1.2 Краткое описание возможностей

Подсистема "Взаимодействие с внешними голосовыми ботами (в т.ч. технологии ИИ) 122" 3.0.3_1 предназначена для реализации взаимодействия с внешними голосовыми ботами в целях обеспечения функционирования типовых сценариев голосовой коммуникации на номере 122 сторонних решений в части интеграции с ЕЦП.МИС.

Сервис "Вызов врача на дом" обеспечивает вызова врача на дом, с возможностью вызвать СМП или НПМ с помощью голосового помощника.

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 Общие сведения

Сервис "Вызов врача на дом" предназначен для взаимодействия Системы с Голосовым помощником для вызова врача на дом с возможностью вызвать СМП или НМП, в том числе для регистрации:

- стандартного вызова врача;
- вызова СМП;
- вызова НМП.

2.1 Описание информационного взаимодействия

Для корректного сохранения данных в подсистеме и последующего обращения к ним дополнительно реализуется обмен SOAP и REST-методами.

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

2.2 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.

2.3 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.

2.4 HTTP ошибки

На каждый запрос к сервису выдается ответ в виде HTTP кода состояния. В случаях ошибок, возникших при выполнении запроса, в теле ответа возвращается код состояния, прикладной код ошибки, сообщение об ошибке.

Общие HTTP ошибки возвращаются HTTP кодами.

- 401 – не передан или неверный apiKey
- 404 – обращение к несуществующему методу

2.4.1 Ошибки на стороне клиента

Ошибки на стороне клиента представлены в таблице 1.

Таблица 1 – Ошибки на стороне клиента

| HTTP-код состояния | Прикладной код ошибки | Описание ошибки/Сообщение об ошибке | Примечание |
|------------------------|-----------------------|---|---|
| 400 Bad Request | | Не заполнены обязательные параметры запроса | http-ответ сервиса в случае, если в формат запроса не соответствует требованиям или однако имеется какая-то логическая ошибка, из-за которой невозможно произвести операцию над ресурсом. |
| 401 Unauthorized | | Требуется аутентификация | http-ответ сервиса в случае, если пользователь не прошел аутентификацию |
| 405 Method Not Allowed | | Метод не поддерживается | http-ответ сервиса в случае, если запрашиваемый метод не поддерживается сервисом |
| 408 Request Timeout | | | http-ответ в случае, если истекло время ожидания сервером передачи от клиента истекло. |

2.4.2 Ошибки на стороне сервера

В случае возникновения ошибки на стороне сервера, при исполнении прикладного метода интеграции, в ответе метода возвращается ошибка с кодом 500.

Имеется возможность получения детальной информации по таким ошибкам.

Для получения детальной информации по ошибкам на стороне сервера, в настройках Web-сервиса должна быть включена отладка:

```
api.yaml
apiErrors:
  debugSave: true
```

Для просмотра подробностей используется метод GET /app/errors (Метод получения детальной информации об ошибке) со значением id ошибки, которую вернул метод, породивший ошибку 500.

2.4.3 Детализация ошибок

Для ошибок на стороне клиента (400) предусмотрена детализация с помощью прикладных кода и текста ошибки.

Для ошибок на стороне сервера (500) предусмотрена детализация с помощью прикладных кода и текста ошибки, а, также, возможность получения подробностей ошибки отдельным методом GET /app/errors (Метод получения детальной информации об ошибке).

3 Авторизация

3.1 Авторизация в PHP API. Авторизация в системе

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.

При авторизации используется метод GET /api/user/login.

Таблица 2 – Сценарий взаимодействия с ИС "Авторизация в PHP API"

| | |
|--|--|
| Интегрируемые системы | <ul style="list-style-type: none">– Система:– модуль АРМ администратора МО– Внешняя ИС |
| Инициатор сценария: | Внешняя ИС |
| Предварительные условия: | Внешней ИС выданы учетные данные для авторизации в Системе |
| Иницирующее событие: | Вызов метода со стороны Внешней ИС |
| Критерий успешности выполнения: | Авторизация выполнена успешно, в результате работы метода получен идентификатор сессии |
| Базовые шаги сценария: | <ol style="list-style-type: none">1. Внешняя ИС направляет запрос на авторизацию в Систему – GET api/user/login (Авторизация пользователя в системе).2. Система принимает запрос и осуществляет поиск пользователя по переданным логину и паролю.3. Система направляет во Внешнюю ИС ответ с результатом поиска логина и пароля. |
| Альтернативные шаги сценария: | <p>3А. При отсутствии логина или пароля, Система направляет информирующий ответ о том, что неверный логин или пароль:</p> <ul style="list-style-type: none">– Ошибка;– Код ошибки –100;– Сообщение об ошибке "Неверный логин или пароль" |

3.2 Авторизация в SWAN API. Метод авторизации пользователя СМП

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.

Используемый метод – **GET /login**.

Таблица 3 – Сценарий взаимодействия с ИС "Авторизация в SWAN API"

| | |
|--|---|
| Интегрируемые системы | <ul style="list-style-type: none">– Система– Внешняя ИС |
| Инициатор сценария: | Внешняя ИС |
| Предварительные условия: | Сервис развернут. Интеграция включена. |
| Иницирующее событие: | Вызов прикладного интеграционного метода со стороны внешней ИС |
| Критерий успешности выполнения: | Авторизация пользователя выполнена. Последующие вызовы интеграционных методов успешны. В случае возникновения ошибок выполнения методов - ошибки возвращаются в описанном формате |
| Базовые шаги сценария: | <ol style="list-style-type: none">1. Внешняя ИС формирует параметры и направляет в ЕЦП запрос GET /rest-api/login (Метод авторизации пользователя) на авторизацию пользователя Системы.2. При возникновении ошибок на стороне сервера, при выполнении методов, внешняя ИС формирует параметры и направляет в Систему запрос GET /app/errors (Метод получения детальной информации об ошибке) на получение детальной информации об ошибке |

4 Описание сервиса "Вызов врача на дом"

Таблица 4 – Описание сценария "Вызов врача на дом"

| | |
|---|---|
| <p>Интегрируемые системы</p> | <ul style="list-style-type: none"> – Система: <ul style="list-style-type: none"> – Подсистема "Неотложная медицинская помощь" (НМП); – Подсистема "Регистратура": <ul style="list-style-type: none"> ▪ модуль "Вызов врача на дом". – Подсистема "Скорая медицинская помощь" (СМП); – Подсистема "Взаимодействие с внешними голосовыми ботами (в т.ч. технологии ИИ) 122". – Голосовой помощник (ГП) |
| <p>Инициатор сценария:</p> | <p>ГП</p> |
| <p>Предварительные условия:</p> | <p>ГП авторизован в РНР API Системы</p> |
| <p>Иницирующее событие:</p> | <p>Пациент совершает звонок в службу с ГП и выбирает опцию "Вызов врача на дом"</p> |
| <p>Критерий успешности выполнения:</p> | <p>В ГП передан идентификатор вызова на дом, созданного в Системе</p> |
| <p>Базовые шаги сценария:</p> | <ol style="list-style-type: none"> 1. Пациент звонит в службу голосового помощника и выбирает опцию "Вызов врача на дом". 2. Голосовой помощник для идентификации пациента просит его назвать единый номер полиса ОМС. 3. Пациент называет свои данные 4. ГП направляет в Систему запрос GET /api/Person/mgetPersonSearch (Поиск человека по атрибутам) с названным пациентом ЕНП ОМС. 5. Система отправляет данные о человеке 6. ГП направляет в Систему запрос GET api/Address (Получение адреса) с полученным идентификатором пациента Person_id. 7. Система возвращает ГП найденный адрес. 8. ГП называет адрес и уточняет у пациента, вызвать ли врача на него 9. ГП узнаёт у пациента симптомы. 10. ГП направляет в Систему запрос POST api/HomeVisit/HomeVisit (Добавление вызова врача на дом). 11. Система создаёт вызов на дом. 12. Система возвращает данные о вызове на дом |
| <p>Альтернативные шаги сценария:</p> | <p>ЕНП ОМС не найден в системе</p> <ol style="list-style-type: none"> 4А. Система направляет в ГП пустой ответ. 5А. ГП спрашивает у пациента, возможна ли запись по адресу. 6А. ГП узнаёт у пациента его личные данные. 7А. ГП направляет в Систему запрос POST api/Person (Создание человека). 8А. Система создаёт человека. 9А. Система возвращает ИД пациента. |

| | |
|--|---|
| | <p>10А. ГП узнаёт у пациента данные о его адресе. 11А. ГП направляет в Систему запрос GET api/AddressSearch (Получение списка параметров адреса). 12А. Система проверяет наличие адреса в системе. 13А. Переход к шагу 9 базового сценария</p> |
| | <p>Адрес не найден в системе 13В. Система возвращает пустой ответ. 14В. ГП переводит пациента на оператора</p> |
| | <p>Пациент хочет вызвать врача на другой адрес 9А. Переход к шагу 10А альтернативного сценария "ЕМП ОМС не найден в системе"</p> |
| | <p>Симптомы пациента соответствуют критериям, по которым должна быть вызвана СМП или НМП 10А. ГП направляет в Систему запрос POST /emergency/EmergencyCallCard (Метод добавления талона вызова). 11А. Система возвращает данные о вызове</p> |

5 Описание методов

5.2 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;
- 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.

5.3 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;

- Н – необязательный реквизит, который может, как присутствовать, так и отсутствовать в элементе. При отсутствии, не передается;
- У – условно–обязательный реквизит. При отсутствии, не передается;
- М – 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

5.4 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.

5.5 Описание методов работы с человеком

5.5.1 Создание человека POST api/Person

Метод предназначен для создания человека.

Алгоритм выполнения метода:

- 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) 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) Если переданная дата рождения больше текущей даты, то ошибка "Некорректное значение в поле PersonBirthDay_BirthDay". Создание человека не происходит.
- е) 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) Если в передаваемом параметре (Person_Sex_id, SocStatus_id, Address_id, PAddress_id, VAddress_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) 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 (Т, О): 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.

Таблица 5 – Параметры метода **POST api/Person**

| | |
|---------------------------|---|
| Входящие параметры | <ul style="list-style-type: none"> – Person_isUnknown (N, H) – признак "Личность неизвестна"; – PersonSurName_SurName (Т, О) – фамилия; – PersonFirName_FirName (Т, H) – имя; – PersonSecName_SecName (Т, H) – отчество; – PersonBirthDay_BirthDay (D, O) – дата рождения; – Person_Sex_id (N, O) – пол; – PersonPhone_Phone (N, H) – телефон; – PersonSnils_Snils (N,H) – СНИЛС; – SocStatus_id (N, O) – 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; |
|---------------------------|---|

| | |
|--------------|---|
| | <ul style="list-style-type: none"> - 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; - fromMobile (N,H) – признак мобильного телефона; - options (S, H) – настройка работы метода. Допустимые форматы: JSON или массива. Параметры настройки: - CheckChecksumSnilsOff (N, H): <ul style="list-style-type: none"> - 1 – отключить проверку контрольной суммы СНИЛС; - 0 или пропущен – проверять контрольную сумму СНИЛС. - CheckDuplicateSnilsOff (N,H): <ul style="list-style-type: none"> - 1 – отключить проверку дублирования СНИЛС; - 0 или пропущен – проверять СНИЛС на дублирование - DuplicateSearch (T, H) – 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,H) – территория страхования; - OrgSMO_id (N,H) – полис выдан; - PolisType_id (N,H) – тип полиса; - PolisFormType_id (N,H) – форма полиса; - Polis_EdNum (T ,H) – 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 | <p>В случае корректной обработки запроса и успешного создания записи в ответе будет прислано:</p> <ul style="list-style-type: none"> - код статуса: 200 - значение параметров: <ul style="list-style-type: none"> - error_code: 0 - data: - Person_id: значение параметра <p>Если в запросе не заполнен хотя бы один из обязательных параметров, то:</p> |

| | |
|-------------------------------|---|
| | <ul style="list-style-type: none"> - код статуса: 200 - значение параметров: <ul style="list-style-type: none"> - error_code: 6 - error_msg: "Отсутствует один из обязательных параметров PersonSurName_SurName, PersonBirthDay_BirthDay, Person_Sex_id, SocStatus_id" <p>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:</p> <ul style="list-style-type: none"> - <=7; - >= возраст на текущую дату <= 18; - >18 возраст на текущую дату <=60; - возраст на текущую дату > 60; |
| <p>Описание ошибок</p> | <ul style="list-style-type: none"> - 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 – |

| | |
|--|---|
| | <p>отображается ошибка "Значение поля <Названия полей через запятую> не соответствует ожидаемому";</p> <ul style="list-style-type: none"> - 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" |
|--|---|

5.5.2 Редактирование данных человека PUT api/Person

Метод предназначен для редактирования информации по человеку.

Таблица 6 – Параметры метода PUT api/Person

| | |
|---------------------------|---|
| Входящие параметры | <ul style="list-style-type: none"> - 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 законно 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

| | |
|--------------|--|
| Ответ | <p style="text-align: center;">пациента) –параметр DeputyKind_id становится обязательным</p> <ul style="list-style-type: none"> – 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 Person_id не найден пациент, то ошибка "Пациент не найден в системе". – Если в любом передаваемом параметре передается значение, не соответствующее ожидаемому формату, например: <ul style="list-style-type: none"> – ожидается числовое значение, а передано текстовое значение; – ожидается параметр, состоящий из 11 символов, а передан параметр, состоящий из большего или меньшего количества символов; <p>то ошибка: "Неверный формат поля <Наименования полей через запятую>". Обновление человека в БД не происходит.</p> <ul style="list-style-type: none"> – 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 Person_id 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 id (Person_Sex_id, SocStatus_id, Address_id, PAddress_id, VAddress_id, Org_id, Post_id), 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) (Person_isUnknown). 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 SocStatus_id), состоящее из цифр. Если результат проверки отрицательный, то выводится ошибка о неверном формате входного параметра. – Правила для параметров: – PersonSurName_SurName (Т, У): ... <p>Если параметр не указан, то он не проверяется.</p> <p>Если параметр указан и пустой, то метод вернёт ошибку: Значения для (PersonSurName_SurName) обязательное и не может быть пустым.</p> |
|--------------|--|

- PersonBirthDay_BirthDay (D, Y): ...

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, Y): ...

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

5.5.3 Поиск человека по атрибутам GET /api/Person/mgetPersonSearch

Мобильный метод предназначен для поиска человека по атрибутам.

Таблица 7 – Параметры метода GET /api/Person/mgetPersonSearch

| | |
|---------------------------|---|
| Входящие параметры | <p>Условия формирования параметров запроса:</p> <ul style="list-style-type: none">– 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):<ul style="list-style-type: none">– 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 предвостра несовершеннолетних– evnpldtipro - если ищем по картам первого этапа профосмотра несовершеннолетних– withlgotonly - только по льготникам– dddispclass2 - ДВН 2– ddorpperiod - периодический осмотр– ddorpssec - если ищем по картам первого этапа детей-сирот– hasopenevnps - если ищем по открытым КВС– ddorp или ddorp_all - сли ищем по регистру детей-сирот– geriatics - 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 объёме– att_1_4 - только прикрепленные (основное или служебное)– att - только прикрепленные– person_register_by_vzn_request - поиск в регистре для заявки ВЗН |
|---------------------------|---|

| | |
|----------------------|--|
| | <ul style="list-style-type: none"> - hms - поиск для регистра главных внештатных специалистов - fmba - только не включенные в регистр ФМБА - dd - если ищем по ДД - encryponly - 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 |
| <p>Пример</p> | <p>Успешный ответ:</p> <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", }] } </pre> |

```
"PersonSecName_SecName": "АЛЕКСЕЕВИЧ",
"Polis_Ser": null,
"PolisFormType_id": null,
"OrgSMO_id": null,
"OMSSprTerr_id": null,
"Polis_endDate": null,
"Person_Snils": "06252245839",
"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"
},
{
"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",
"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> }</pre> |
| Ответ | <p>Логика формирования параметров ответа:</p> <p>Person_IsBDZ. Значение по умолчанию:</p> <ul style="list-style-type: none"> - 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: <ul style="list-style-type: none"> - 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: <ul style="list-style-type: none"> - Если person.Person_IsInErz = 1, то 'orange' - Если person.Person_IsInErz = 2, то 'true' - Иначе 'false' - Для Казахстана - <ul style="list-style-type: none"> - Если person.Person_IsInErz = 1, то 'red' - Если person.Person_IsInErz = 2, то 'true'Иначе 'false' |
| Ошибки | <p>Если не указаны параметры запроса:</p> <pre> { "success": false, "error_msg": "Не указан ни один фильтр. Для поиска заполните хотя бы один параметр в блоке фильтров.", "error_code": 6 }</pre> |

5.6 Методы сервиса "Вызов врача на дом"

5.6.2 Добавление вызова врача на дом POST api/HomeVisit/HomeVisit

Таблица 8 – Параметры метода POST api/HomeVisit/HomeVisit

| | |
|---------------------------|---|
| Входящие параметры | <ul style="list-style-type: none">– 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 |
|---------------------------|---|

| | |
|---------------------|--|
| | <p>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";</p> <ul style="list-style-type: none"> - KLAAdr_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) |
| <p>Ответ</p> | <p>Успешный ответ</p> <p>Код ошибки – 0</p> <ul style="list-style-type: none"> - 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) 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) 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": 10101000000005,
        "LpuRegion_Name": "777",
        "MedStaffFact_id": "140101000001873",
        "Description": "Врач не обслуживает участок, потому что указанное место работы в параметре MedStaffFact_id [140101000000040] не соответствует месту работы врача."
      }
    ]
  ],
  "data": {
    "HomeVisit_id": "140101000000737",
    "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;

| | |
|----------------------|--|
| | <ul style="list-style-type: none"> – 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). <p>Успешный ответ:</p> <p>Код ошибки – 0</p> <ul style="list-style-type: none"> – 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) |
| <p>Ошибки</p> | <p>Код ошибки – 6</p> <ul style="list-style-type: none"> – 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.

| | |
|----------------------|---|
| | <ul style="list-style-type: none"> – 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. |
| <p>Пример</p> | <p>Запрос:</p> <p>POST</p> <p>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</p> <p>Ответ:</p> <p>Ошибка</p> <pre>{ "error_code": 0, "data": [{ "HomeVisit_id": "590101000000223" }] }</pre> |

| | |
|--|--------|
| |] } |
|--|--------|

5.6.3 Получение адреса GET api/Address

Таблица 9 – Параметры метода GET api/Address

| | |
|---------------------------|---|
| Входящие параметры | <ul style="list-style-type: none"> – 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 (T, H) – GUID территории (города, населенного пункта и т.п.) по ФИАС; – Street_AOGUID (T, H) – GUID улицы по ФИАС |
| Ответ | <p>Если не задан ни Person_id, ни Address_id, то ошибка (хотя бы один из параметров должен быть задан).</p> <p>Если не найдено ни одной записи, то ответ "0".</p> <p>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)".</p> <p>Если хотя бы одна запись найдена, то успешный ответ – список адресов с информацией:</p> <ul style="list-style-type: none"> – 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; |

| | |
|---------------|---|
| | <ul style="list-style-type: none"> – KLStreet_id (N, O) – улица; – Address_House (N, O) – дом; – Address_Corpus (N, H) – корпус; – Address_Flat (N, H) – квартира; – Address_Address (T, H) – 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 (T, O) – наименование страны; – KLRgn_name (T, Y) – наименование региона; – KLSubRgn_name (T, Y) – наименование района; – KLCity_name (T, Y) – наименование города; – KLTown_name (T, Y) – наименование населенного пункта; – KLStreet_name (T, Y) – наименование улицы. <p>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:</p> <ul style="list-style-type: none"> – код состояния: 200 – значение параметров: <ul style="list-style-type: none"> – error_code: 23001 – error_msg: Отсутствует обязательный параметр (наименование параметра) |
| Пример | <pre>{ "error_code": 0,</pre> |

```
"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, ТРАКТОРНАЯ УЛ, д 56, кв 6",  
    "Address_insDT": "2020-10-09",  
    "AoidArea": "1696D568-7A59-48D6-A81D-B982C9C3AD0F",  
    "AoidStreet": "8557F0BF-A01F-4411-B48F-B347312ED69F",
```

| | |
|--|---|
| | <pre> "KLAreaType_id": "1", "AOGUIDArea": "A309E4CE-2F36-4106-B1CA-53E0F48A6D95", "AOGUIDStreet": "8CB4A0FE-1BCC-4BD6-B4A8-941DF5E8F61B", "Address_insDatetime": "2020-10-09 12:10:58" }] } </pre> |
|--|---|

5.6.4 Получение списка параметров адреса GET api/AddressSearch

Метод возвращает список параметров адреса.

Таблица 10 – Параметры метода GET api/AddressSearch

| | |
|---------------------------|---|
| Входящие параметры | <ul style="list-style-type: none"> – address (T,O) – 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. <p>1 – Метод вернёт дополнительную информацию результата анализа поиска адреса.</p> |
| Ответ | <ul style="list-style-type: none"> – Country_id (N) – идентификатор страны. – Country_Code (T) – Код страны (ISO). – Country_Name (T) – Наименование страны. – Area_id (N) – идентификатор области/края/республики и т.д. – Area_Name (T) – Наименование области/края/республики и т.д. – Area_Type (T) – Тип области/края/республики и т.д. – Area_KLADR_Code (T) – Код КЛАДР области/края/республики и т.д. – Area_Postal_Index (T) – Почтовый индекс |

| | |
|--|--|
| | <p>области/края/республики и т.д.</p> <ul style="list-style-type: none"> – Area_AOGUID (T) – идентификатор адресного объекта по ФИАС (области/края/республики и т.д.). – SubRegion_id (N) – идентификатор района. – SubRegion_Name (T) – Наименование района. – SubRegion_Type (T) – 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. |
|--|--|

| | |
|--|---|
| | <p>Правила:</p> <ul style="list-style-type: none"> – 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.KLSocг 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. |
|--|---|

5.6.5 Метод добавления талона вызова POST /emergency/EmergencyCallCard

Метод предназначен для добавления талона вызова

Таблица 11 – Параметры метода **POST /emergency/EmergencyCallCard**

| | |
|---------------------------|---|
| Входящие параметры | <ul style="list-style-type: none"> – CmpCallCard – талон вызова: <ul style="list-style-type: none"> – 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); |
|---------------------------|---|

| | |
|--|---|
| | <ul style="list-style-type: none"> – house (Т (30)) – номер дома адреса; – корпус (Т (5)) – номер корпуса адреса; – flat (Т (5)) – номер квартиры; – room (Т (5)) – номер комнаты; – entrance (Т (200)) – номер подъезда. Размерность параметра 200 символов; – floor (Т (200)) – номер этажа. Размерность параметра 200 символов; – entranceCode (Т (200)) – код домофона. Размерность параметра 200 символов; – phone (Т (20)) – номер телефона; – comment (Т (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 (Т (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 (Т (30)) – фамилия пациента; – firname (Т (30)) – имя; – secname (Т (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 (Т (10)) – серия полиса ОМС; – polis.number (Т (30)) – номер полиса ОМС; – polis.federalNumber (Т (16)) – единый номер полиса ОМС; – whoCallType.id (long) – вид вызывающего помощь; – whoCall (Т (30)) – кто вызывает помощь; – callType.id (long) – тип вызова (CmpCallType); – callProfile.id (long) – профиль вызова (CmpProfile); – urgency (N) – срочность вызова; – acceptTime (Т (DT) (23), Y) – дата и время получения вызова; – endingTime (Т (DT) (23), Y) – дата и время окончания вызова; – arrivalPlanTime (Т (DT) (23)) – плановое время прибытия на место вызова; |
|--|---|

| | |
|--|---|
| | <ul style="list-style-type: none"> - 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; - medstafTeam.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), N) – текстовое описание адреса вызова; - commonState.id (long) – идентификатор степени тяжести состояния; |
|--|---|

| | |
|---------------------|---|
| | <ul style="list-style-type: none"> – 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), Н) – серия и номер документа удостоверяющего личность; – document.orgDepText (string(1000), Н) – организация, выдавшая документ удостоверяющий личность; – document.begDate (datetime, Н) – дата выдачи документа удостоверяющего личность; – uaddress (string (1000), Н) – адрес места жительства пациента; – accidentCircumstance (string(512), Н) – обстоятельства несчастного случая; – SmpEmTeamDuty: <ul style="list-style-type: none"> – 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) |
| <p>Ответ</p> | <p>Пример ответа:</p> <pre>{ "errorCode": 0, "errorMessage": "OK", "rowCount": 1, "responseData": { "errorCode": 0, "errorMessage": "OK", "id": 590910000083170, "guid": null,</pre> |

```
"stringId": null,  
"ids": null,  
"version": null,  
"versionOld": null,  
"complex": false,  
"success": true  
},  
"success": true  
}
```

6 Аварийные ситуации

6.1 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 характеризуемое 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.

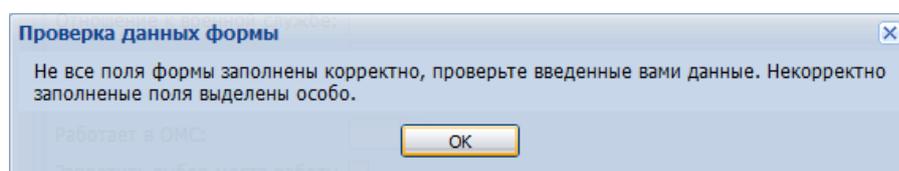
6.2 Действия в случае несоблюдения условий выполнения технологического процесса

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.