

**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

## 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	Общие сведения .....	6
2.1	Описание информационного взаимодействия.....	6
2.2	Схема взаимодействия .....	6
2.3	Кэширование запросов.....	6
2.4	HTTP ошибки.....	7
2.4.1	Ошибки на стороне клиента.....	7
2.4.2	Ошибки на стороне сервера .....	7
2.4.3	Детализация ошибок .....	8
3	Авторизация .....	9
3.1	Авторизация в PHP API. Авторизация в системе.....	9
3.2	Авторизация в SWAN API. Метод авторизации пользователя СМП.....	10
4	Описание сценариев .....	11
4.1	Сервис "Запись к врачу" .....	11
4.2	Сервис "Подтверждение записи на прием к врачу с возможностью отмены и переноса (исходящий обзвон)" .....	13
4.3	Сервис "Вызов врача на дом" .....	15
4.4	Сервис "Запись на вакцинацию" .....	16
5	Описание методов.....	18
5.1	Общие параметры для всех методов .....	18
5.2	Описание параметров метода.....	18
5.3	Описание общих кодов ошибок.....	19
5.4	Описание методов работы с человеком .....	19
7.1.1.	Получение списка физических лиц, по ключевым параметрам GET api/PersonList.....	19
7.1.2.	Получение информации по человеку GET api/Person .....	20
7.1.3.	Создание человека POST api/Person.....	22
7.1.4.	Редактирование данных человека PUT api/Person .....	27

7.1.5.	Получение записей на прием к врачу	GET	api/TimeTableGraf/TimeTableGrafByMedStaffFact .....	30
7.1.6.	Получение прикреплений пациента	GET	api/PersonAttach .....	31
7.1.7.	Получение основного прикрепления пациента	GET	api/PersonMainAttach .....	32
5.5	Общие методы .....			35
5.5.1	Получение элементов справочника	GET	api/Refbook .....	35
5.5.2	Поиск человека по атрибутам	GET	/api/Person/mgetPersonSearch .....	37
5.5.3	Работа с направлениями .....			42
5.5.3.1	Создание направления	POST	api/EvnDirection .....	42
5.6	Методы работы с листами ожидания .....			51
5.6.1	Добавление записи в лист ожидания	POST	api/EvnQueue/EvnQueue .....	51
5.6.2	Получение листа ожидания	GET	api/EvnQueue/EvnQueue .....	52
5.7	Методы сервиса "Запись к врачу" .....			53
5.7.1	Получение должности по наименованию	GET	api/PostByName .....	53
5.7.2	Получение МО по идентификатору	GET	api/MOById .....	53
5.7.3	Получение информации о докторе	GET	/api/doctors/MedStaffFact_id .....	53
5.7.4	Получение списка специальностей в МО	GET	api/MedSpecOms/MedSpecOmsByMO .....	57
5.7.5	Получение списка мест работы по специальности и МО, или по должности и МО	GET	api/MedStaffFact/MedStaffFactByMO .....	57
5.7.6	Получение свободных дат приема	GET	api/TimeTableGraf/TimeTableGrafFreeDate .....	60
5.7.7	Получение свободного времени приема	GET	api/TimeTableGraf/TimeTableGrafFreeTime .....	60
5.7.8	Запись пациента на прием	POST	api/TimeTableGraf/TimeTableGrafWrite .....	62
5.8	Методы сервиса "Подтверждение записи на прием к врачу с возможностью отмены и переноса (исходящий обзвон)" .....			63
5.8.1	Получение списка прошедших и предстоящих занятых бирок в МО за заданный временной промежуток	GET	api/TimeTableList .....	63
5.8.2	Получение списка предстоящих приемов пациента в МО	GET	api/TimeTableListbyPatient .....	65
5.8.3	Изменение статуса записи на прием	PUT	api/TimeTableGraf/TimeTableGrafStatus .....	67
5.9	Методы сервиса "Вызов врача на дом" .....			67
5.9.1	Добавление вызова врача на дом	POST	api/HomeVisit/HomeVisit .....	67
5.9.2	Получение адреса	GET	api/Address .....	76
5.9.3	Получение списка параметров адреса	GET	api/AddressSearch .....	79
5.9.4	Метод добавления талона вызова	POST	/emergency/EmergencyCallCard .....	81

5.10	Методы сервиса "Запись на вакцинацию" .....	86
5.10.1	Получение расписания на вакцинацию в МО прикрепления пациента и ближайшие MO GET api/VaccinationTimeTableMedService .....	86
5.10.2	Запись пациента на приём службы POST api/TimeTableMedService/TimeTableMedServiceWrite .....	87
6	Аварийные ситуации .....	89
6.1	Описание аварийных ситуаций .....	89
6.2	Действия в случае несоблюдения условий выполнения технологического процесса ..	90

# **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\_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.

Подсистема поддерживает работу следующих сервисов:

- сервис "Запись к врачу";
- сервис "Подтверждение записи на прием к врачу с возможностью отмены и переноса (исходящий обзвон)";
- сервис "Запись на вакцинацию";
- сервис "вызов врача на дом".

## **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

Подсистема реализует прием данных от внешних информационных систем (платформ) для создания голосовых ботов, в т. ч. обладающих технологиями искусственного интеллекта, по данным паспортов МО, структуре МО, расписанию, документам в соответствии с федеральным форматом. Взаимодействие осуществляется в сетевом контуре размещения МИС. Позволяет осуществлять информационный обмен и реализацию типовых сценариев здравоохранения.

Подсистема поддерживает работу следующих сервисов:

- сервис "Запись к врачу" – создание записи на приём к врачу (без направления);
- сервис "Подтверждение записи на прием к врачу с возможностью отмены и переноса (исходящий обзвон)" – подтверждение записи на прием к врачу путём исходящего обзвона:
  - отмена записи;
  - перенос записи.
- сервис "Запись на вакцинацию" – создание записи на вакцинацию;
- сервис "вызов врача на дом":
  - стандартный вызов врача;
  - вызов СМП;
  - вызов 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"

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

### 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"

<b>Интегрируемые системы</b>	<ul style="list-style-type: none"><li>– Система</li><li>– Внешняя ИС</li></ul>
<b>Инициатор сценария:</b>	Внешняя ИС
<b>Предварительные условия:</b>	Сервис развернут. Интеграция включена.
<b>Иницирующее событие:</b>	Вызов прикладного интеграционного метода со стороны внешней ИС
<b>Критерий успешности выполнения:</b>	Авторизация пользователя выполнена. Последующие вызовы интеграционных методов успешны. В случае возникновения ошибок выполнения методов - ошибки возвращаются в описанном формате
<b>Базовые шаги сценария:</b>	<ol style="list-style-type: none"><li>1. Внешняя ИС формирует параметры и направляет в ЕЦП запрос GET /rest-api/login (Метод авторизации пользователя) на авторизацию пользователя Системы.</li><li>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</li></ol>

## 4 Описание сценариев

### 4.1 Сервис "Запись к врачу"

Таблица 4 – Описание сценария "Запись на прием к врачу (без направления)"

<b>Интегрируемые системы</b>	<ul style="list-style-type: none"> <li>– Система: <ul style="list-style-type: none"> <li>– Подсистема "Регистратура";</li> <li>– Подсистема "Взаимодействие с внешними голосовыми ботами (в т.ч. технологии ИИ) 122".</li> </ul> </li> <li>– Голосовой помощник (ГП)</li> </ul>
<b>Инициатор сценария:</b>	ГП
<b>Предварительные условия:</b>	<ul style="list-style-type: none"> <li>– ГП авторизован в RHP API Системы;</li> <li>– синхронизированы справочники специальностей методом GET api/Refbook (Получение элементов справочника);</li> <li>– наименования и идентификаторы специальностей сопоставлены методом GET api/PostByName (Получение должности по наименованию)</li> </ul>
<b>Иницирующее событие:</b>	Пациент совершает звонок в службу с ГП и выбирает опцию "Запись на приём к врачу"
<b>Критерий успешности выполнения:</b>	В ГП передан идентификатор записи на приём к врачу, созданной в Системе
<b>Базовые шаги сценария:</b>	<ol style="list-style-type: none"> <li>1. Пациент звонит в службу голосового помощника и выбирает опцию "Запись на приём к врачу".</li> <li>2. ГП для идентификации пациента просит его назвать единый номер полиса ОМС.</li> <li>3. Пациент называет свои данные</li> <li>4. ГП направляет в Систему запрос GET /api/Person/mgetPersonSearch (Поиск человека по атрибутам) с названным пациентом ЕНП ОМС.</li> <li>5. Система возвращает данные о человеке.</li> <li>6. ГП направляет в Систему запрос GET api/PersonMainAttach (Получение основного прикрепления пациента).</li> <li>7. Система возвращает ГП информацию о прикреплении пациента.</li> <li>8. ГП направляет в Систему запрос GET api/МОById (Получение МО по идентификатору).</li> <li>9. Система возвращает ГП информацию о МО прикрепления пациента</li> <li>10. ГП спрашивает у пациента специальность врача, к которому необходимо оформить запись.</li> <li>11. ГП направляет в Систему запрос GET api/TimeTableListbyPatient (Получение списка прошедших и предстоящих бирок пациента в МО) для проверки того, записан ли уже пациент к врачу данной специальности.</li> <li>12. Система возвращает все записи пациента.</li> <li>13. ГП направляет запрос при помощи метода GET</li> </ol>

	<p>api/MedSpecOms/MedSpecOmsByMO для получения кодов специальностей в МО.</p> <p>14. Система отправляет в ответ список специальностей МО</p> <p>15. ГП уточняет у пациента, подойдёт ли ему запись в данную МО.</p> <p>16. ГП направляет в Систему запрос GET api/MedStaffFact/MedStaffFactByMO (Получение списка мест работы по специальности и МО, или по должности и МО).</p> <p>17. Система возвращает список врачей, к которым можно записаться.</p> <p>18. ГП спрашивает у пациента, хочет ли он записаться к конкретному врачу.</p> <p>19. ГП направляет в Систему запрос GET api/TimeTableGraf/TimeTableGrafFreeDate (Получение свободных дат приема).</p> <p>20. Система возвращает свободные даты приёма.</p> <p>21. ГП направляет в Систему запрос GET api/TimeTableGraf/TimeTableGrafFreeTime (Получение свободного времени приема).</p> <p>22. Система. возвращает свободные бирки приёма</p> <p>23. ГП направляет в Систему запрос POST api/TimeTableGraf/TimeTableGrafWrite (Запись пациента на прием).</p> <p>24. Система создаёт запись на приём.</p> <p>25. Система возвращает данные о приёме</p>
<p><b>Альтернативные шаги сценария:</b></p>	<p><b>ЕИП ОМС не найден в системе</b></p> <p><b>5А.</b> Система направляет в ГП пустой ответ.</p> <p><b>6А.</b> ГП переводит пациента на оператора</p> <hr/> <p><b>Можно записаться не по месту основного прикрепления</b></p> <p>8А. ГП направляет в Систему запрос GET api/PersonAttach (Метод получения прикреплений пациента).</p> <p>9А. Переход к шагу 9 основного сценария</p> <hr/> <p><b>У пациента уже существует запись к врачу</b></p> <p>15А. ГП уточняет у пациента, удалить ли существующую запись.</p> <p>16А. ГП направляет в Систему запрос PUT api/TimeTableGraf/TimeTableGrafStatus (Изменение статуса записи на прием) с параметром EvnStatus_id=12.</p> <p>17А. Система удаляет запись на приём.</p> <p>18А. ГП спрашивает у пациента, создать ли новую запись.</p> <p>19А. Переход к шагу 17 основного сценария</p> <hr/> <p><b>Пациент хочет записаться к конкретному врачу</b></p> <p><b>18А.</b> ГП перечисляет врачей, к которым можно записаться.</p> <p><b>19А.</b> Переход к шагу 19 основного сценария</p> <hr/> <p><b>Пациенту не подходит ни одна из существующих бирок</b></p> <p>22А. ГП спрашивает у пациента, записать ли его в лист ожидания.</p> <p>23А. ГП спрашивает у пациента, записать ли его к конкретному врачу.</p> <p>24А. ГП направляет в Систему запрос POST api/EvnQueue/EvnQueue (Добавление записи в лист ожидания) с указанием конкретного врача.</p>

	25А. Система направляет в ГП данные о записи в лист ожидания
	<b>Пациент хочет записаться в лист ожидания к любому врачу по специальности</b>
	24В. ГП направляет в Систему запрос POST api/EvnQueue/EvnQueue (Добавление записи в лист ожидания) с указанием специальности врача.
	25В. Система направляет в ГП данные о записи в лист ожидания

#### 4.2 Сервис "Подтверждение записи на прием к врачу с возможностью отмены и переноса (исходящий обзвон)"

Таблица 5 – Описание сценария "Подтверждение записи на прием к врачу (исходящий обзвон)"

<b>Интегрируемые системы</b>	<ul style="list-style-type: none"> <li>– Система:</li> <li>– Подсистема "Регистратура";</li> <li>– Подсистема "Взаимодействие с внешними голосовыми ботами (в т.ч. технологии ИИ) 122".</li> <li>– Голосовой помощник (ГП)</li> </ul>
<b>Инициатор сценария:</b>	Голосовой помощник (ГП)
<b>Предварительные условия:</b>	<ul style="list-style-type: none"> <li>– ГП авторизован в RHP API Системы;</li> <li>– в БД создана запись к врачу со следующими данными: <ul style="list-style-type: none"> <li>– идентификатор МО;</li> <li>– наименование МО;</li> <li>– ИД пациента;</li> <li>– фамилия пациента;</li> <li>– имя пациента;</li> <li>– отчество пациента;</li> <li>– номер телефона пациента;</li> <li>– идентификатор бирки</li> </ul> </li> </ul>
<b>Иницирующее событие:</b>	ГП направляет в Систему запрос GET api/TimeTableList (Получение списка прошедших и предстоящих занятых бирок в ЛПУ за заданный временной промежуток) для формирования списка для обзвона
<b>Критерий успешности выполнения:</b>	<p>Данные бирки сохранены в соответствии с действиями пациента</p> <p>Варианты:</p> <ul style="list-style-type: none"> <li>– если пациент подтверждает запись на приём, данные остаются прежними;</li> <li>– если пациент отменяет запись, бирка освобождается;</li> <li>– если пациент отменяет запись или переносит её: <ul style="list-style-type: none"> <li>– создаётся новая запись на приём;</li> <li>– старая бирка освобождается</li> </ul> </li> </ul>

<b>Базовые шаги сценария:</b>	<p>1. ГП направляет в Систему запрос GET api/TimeTableList (Получение списка прошедших и предстоящих занятых бирок в ЛПУ за заданный временной промежуток) для формирования списка для обзвона.</p> <p>2. Система возвращает бирки МО</p> <p>3. ГП звонит пациенту и уточняет, сможет ли он посетить врача в указанное время.</p> <p>4. Данные бирки остаются неизменными</p>
<b>Альтернативные шаги сценария:</b>	<p><b>Пациент отменяет приём</b></p> <p>4А. ГП направляет в Систему запрос PUT api/TimeTableGraf/TimeTableGrafStatus (Изменение статуса записи на прием) с параметром EvnStatus_id=12.</p> <p>5А. Система освобождает бирку</p> <p><b>Пациент переносит приём</b></p> <p>4Б. ГП направляет в Систему запрос GET api/MedStaffFact/MedStaffFactByМО (Получение списка мест работы по специальности и МО, или по должности и МО).</p> <p>5Б. Система возвращает список бирок врача.</p> <p>6Б. ГП направляет в Систему запрос GET api/TimeTableGraf/TimeTableGrafFreeDate (Получение свободных дат приема).</p> <p>7Б. Система возвращает свободные даты приёма.</p> <p>8Б. ГП направляет в Систему запрос GET api/TimeTableGraf/TimeTableGrafFreeTime (Получение свободного времени приема).</p> <p>9Б. Система возвращает свободные бирки приёма.</p> <p>10Б. ГП направляет в Систему запрос POST api/TimeTableGraf/TimeTableGrafWrite (Запись пациента на прием).</p> <p>11Б. Система создаёт запись на приём.</p> <p>12Б. Система возвращает данные о приёме.</p> <p>13Б. ГП направляет в Систему запрос PUT api/TimeTableGraf/TimeTableGrafStatus (Изменение статуса записи на прием) с параметром EvnStatus_id=12.</p> <p>14Б. Система освобождает бирку</p>

### 4.3 Сервис "Вызов врача на дом"

Таблица 6 – Описание сценария "Вызов врача на дом"

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

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

#### 4.4 Сервис "Запись на вакцинацию"

Таблица 7 – Описание сценария "Запись на вакцинацию"

<b>Интегрируемые системы</b>	<ul style="list-style-type: none"> <li>– Система: <ul style="list-style-type: none"> <li>– Подсистема "Регистратура";</li> <li>– Подсистема "Взаимодействие с внешними голосовыми ботами (в т.ч. технологии ИИ) 122";</li> <li>– Подсистема "Иммунопрофилактика".</li> </ul> </li> <li>– Голосовой помощник (ГП)</li> </ul>
<b>Инициатор сценария:</b>	ГП
<b>Предварительные условия:</b>	ГП авторизован в РНР API Системы
<b>Иницирующее событие:</b>	Пациент совершает звонок в службу с ГП и выбирает опцию "Запись на вакцинацию"
<b>Критерий успешности выполнения:</b>	В ГП передан идентификатор записи на вакцинацию, созданной в Системе
<b>Базовые шаги сценария:</b>	<ol style="list-style-type: none"> <li>1. Пациент звонит в службу голосового помощника и выбирает опцию "Запись на вакцинацию".</li> <li>2. Голосовой помощник для идентификации пациента просит его назвать единый номер полиса ОМС.</li> <li>3. Пациент называет свои данные</li> <li>4. ГП направляет в Систему запрос GET /api/Person/mgetPersonSearch (Поиск человека по атрибутам). с названным пациентом ЕНП ОМС.</li> <li>5. Система отправляет данные о человеке</li> <li>6. ГП направляет в Систему запрос GET api/VaccinationTimeTableMedService (Получение списка бирок на вакцинацию в ближайшем к месту прикрепления ЛПУ).</li> <li>7. Система возвращает ГП информацию о бирках в МО</li> </ol>

	<p>прикрепления.</p> <p>8. ГП называет адрес и уточняет у пациента, будет ли он проходить вакцинацию там</p> <p>9. Пациент выбирает дату и время вакцинации.</p> <p>10. ГП направляет в Систему запрос POST api/TimeTableMedService/TimeTableMedServiceWrite (Запись пациента на приём службы).</p> <p>11. Система создаёт талон на вакцинацию.</p> <p>12. Система возвращает в ответе метода идентификатор талона</p>
<p><b>Альтернативные шаги сценария:</b></p>	<p><b>ЕНП ОМС не найден в системе</b></p> <p>5А. Система направляет в ГП пустой ответ.</p> <p>6А. ГП переводит пациента на оператора</p>
	<p><b>Пациент хочет пройти вакцинацию по другому адресу</b></p> <p>9А. ГП узнаёт у пациента данные о его адресе</p> <p>10А. ГП направляет в ЕЦП запрос GET api/AddressSearch (Получение списка параметров адреса).</p> <p>11А. Система проверяет наличие адреса в системе.</p> <p>12А1. Переход к шагу 9 базового сценария</p>
	<p><b>Адрес не найден в системе</b></p> <p>12В. Система возвращает пустой ответ.</p> <p>13В. ГП переводит пациента на оператора</p>

## 5 Описание методов

### 5.1 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.

### 5.2 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

### 5.3 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.

### 5.4 Описание методов работы с человеком

#### 7.1.1. 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).

Таблица 8 – Параметры метода GET api/PersonList

<b>Входящие параметры</b>	<ul style="list-style-type: none"> <li>– PersonSurName_SurName (T[50], O) – фамилия;</li> <li>– PersonFirName_FirName (T[50], H) – имя;</li> <li>– PersonSecName_SecName (T[50], H) – отчество;</li> <li>– PersonBirthDay_BirthDay (D, O) – дата рождения;</li> <li>– PersonSnils_Snils (N[11],H) – СНИЛС;</li> <li>– Polis_Ser (T[10], H) – серия полиса;</li> <li>– Polis_Num (N[15], H) – номер полиса</li> </ul>
<b>Ответ</b>	<p>Успешный ответ – список пациентов с указанием следующих параметров:</p> <ul style="list-style-type: none"> <li>– Person_id (N, O) – идентификатор пациента;</li> </ul>

	<ul style="list-style-type: none"> <li>– PersonSurName_SurName (T[50], O) – фамилия;</li> <li>– PersonFirName_FirName (T[50], H) – имя;</li> <li>– PersonSecName_SecName (T[50], H) – отчество;</li> <li>– PersonBirthDay_BirthDay (D, O) – дата рождения;</li> <li>– PersonSnils_Snils (N[11],H) – СНИЛС;</li> <li>– Polis_id (N, H) – полис: <ul style="list-style-type: none"> <li>– PolisType_id (N, O) – тип полиса;</li> <li>– Polis_Ser (T[10], H) – серия полиса;</li> <li>– Polis_Num (N[15], O) – номер полиса</li> </ul> </li> </ul>
--	---

### 7.1.2. Получение информации по человеку GET api/Person

Метод предназначен для получения информации по пациенту.

Таблица 9 – Параметры метода GET api/Person

<b>Входящие параметры</b>	<ul style="list-style-type: none"> <li>– Person_id (N, H) – идентификатор пациента;</li> <li>– PersonSurName_SurName (T[50], H) – фамилия;</li> <li>– PersonFirName_FirName (T[50], H) – имя;</li> <li>– PersonSecName_SecName (T[50], H) – отчество;</li> <li>– PersonBirthDay_BirthDay (D, H) – дата рождения;</li> <li>– PersonSnils_Snils (N[11], H) – СНИЛС;</li> <li>– Polis_Ser (T[10], H) – серия полиса;</li> <li>– Polis_Num (N[15], H) – номер полиса;</li> <li>– Lpu_id (N,H) – идентификатор медицинской организации (МО). Если параметр не указан, то метод вернёт данные для всех медицинских организаций текущего региона;</li> <li>– KLCountry_id (N,H) – идентификатор страны гражданства;</li> <li>– Document_IsTwoNation (N,H) – признак "Двойное гражданство"; параметр принимает значения "0" – нет, "1" – да;</li> <li>– allFields (boolean, H) – все поля. Если передан параметр allFields или ResponseFull, то в ответ выводятся вся информация по человеку;</li> <li>– ResponseFull (boolean, H) – полный ответ. Если передан параметр allFields или ResponseFull, то в ответ выводятся вся информация по человеку;</li> <li>– OtherRegion_MoAttach (boolean, H) – пациенты, прикрепленные к МО другого региона</li> </ul>
<b>Ответ</b>	<p>При запуске метода проводится проверка принадлежности пользователя к учётной записи типа "Вендор API". В случае положительного ответа принадлежность пользователя к МО не проверяется.</p> <ul style="list-style-type: none"> <li>– Если не передан ни один параметр, то ошибка: <ul style="list-style-type: none"> <li>– код состояния: 200</li> <li>– значение параметров: <ul style="list-style-type: none"> <li>– error_code:6</li> <li>– error_msg:"В запросе должен быть хотя бы один</li> </ul> </li> </ul> </li> </ul>

	<p style="text-align: center;">параметр."</p> <ul style="list-style-type: none"> <li>- Если не найдено ни одной записи: <ul style="list-style-type: none"> <li>- код состояния: 200</li> <li>- значение параметров: <ul style="list-style-type: none"> <li>- error_code: 0</li> <li>- data: []</li> </ul> </li> </ul> </li> <li>- Если при обработке запроса обнаружено, что в параметре Polis_Num введено значение, не удовлетворяющее контролю контрольного разряда: <ul style="list-style-type: none"> <li>- код состояния: 200;</li> <li>- значение параметров: <ul style="list-style-type: none"> <li>- error_code:6;</li> <li>- error_msg:"Номер ЕНП заполнен не верно".</li> </ul> </li> </ul> </li> <li>- Если хотя бы одна запись найдена, то успешный ответ – список людей с информацией: <ul style="list-style-type: none"> <li>- Person_id (N, O) – идентификатор пациента;</li> <li>- PersonSurName_SurName (T[50], O) – фамилия;</li> <li>- PersonFirName_FirName (T[50], H) – имя;</li> <li>- PersonSecName_SecName (T[50], H) – отчество;</li> <li>- PersonBirthDay_BirthDay (D, H) – дата рождения;</li> <li>- Person_Sex_id (N, H) – пол;</li> <li>- PersonPhone_Phone (T, H) – телефон;</li> <li>- PersonSnils_Snils (N[11],H) – СНИЛС;</li> <li>- PersonInn_Inn (N, H) – ИНН пациента;</li> <li>- SocStatus_id (N, H) – социальный статус;</li> <li>- SocStatusFed_code (T[20]) – социальный статус по коду справочника Минздрава;</li> <li>- Person_pid (N, H) – идентификатор представителя пациента;</li> <li>- ParPersonSecName_SecName (T[50]) – отчество представителя пациента;</li> <li>- DeputyKind_id (N, H) – статус представителя;</li> <li>- UAddress_id (N, H) – адрес регистрации;</li> <li>- UAddress_Address (T[300]) – адрес регистрации строкой;</li> <li>- PAddress_id (N, H) – адрес проживания;</li> <li>- PAddress_Address (T[300]) – адрес проживания строкой;</li> <li>- Org_id (N, H) – место работы;</li> <li>- BAddress_id (N, H) – адрес рождения;</li> <li>- Org_id (N, H) – место работы;</li> <li>- Post_id (N, H) – должность;</li> <li>- BDZ_guid (N, H) – GUID по БДЗ;</li> <li>- BDZ_id (N, H) – идентификатор по БДЗ;</li> <li>- LegalStatusVZN_id (N) – идентификатор правового статуса нерезидента</li> <li>- LegalStatusVZN_Name (T) – наименование правового статуса нерезидента;</li> <li>- LegalStatusVZN_pid (N) – идентификатор базового правового статуса нерезидента;</li> <li>- LegalStatusVZN_pName (T) – наименование базового правового статуса нерезидента;</li> <li>- Lpu_id (N) – идентификатор медицинской организации</li> </ul> </li> </ul>
--	--

	<p>прикрепления.</p> <ul style="list-style-type: none"> <li>– KLCountry_id (N,H) – идентификатор страны гражданства;</li> <li>– Document_IsTwoNation (N,H) – признак "Двойное гражданство"; параметр принимает значения "0" – нет, "1" – да;</li> <li>– Employment_id (N, H) – идентификатор занятости пациента;</li> <li>– FamilyStatus_id (N,H) – идентификатор семейного положения пациента;</li> <li>– PersonFamilyStatus_IsMarried (N,H) – идентификатор типа нахождения в зарегистрированном браке;</li> <li>– DocumentAuthority_id (N,H) – идентификатор типа документа законного представителя;</li> <li>– DocumentDeputy_Ser (T, H) – серия документа законного представителя;</li> <li>– DocumentDeputy_Num (T, H) – номер документа законного представителя;</li> <li>– DocumentDeputy_Issue (T, H) – кем выдан документ законного представителя;</li> <li>– DocumentDeputy_begDate (D,H) – дата выдачи документа законного представителя;</li> <li>– deputyOrg_id (N, H) – идентификатор организации, являющейся законным представителем пациента</li> </ul>
--	--

### 7.1.3. 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) 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 10 – Параметры метода **POST api/Person**

<b>Входящие параметры</b>	<ul style="list-style-type: none"> <li>– Person_isUnknown (N, H) – признак "Личность неизвестна";</li> <li>– PersonSurName_SurName (Т, О) – фамилия;</li> <li>– PersonFirName_FirName (Т, H) – имя;</li> <li>– PersonSecName_SecName (Т, H) – отчество;</li> <li>– PersonBirthDay_BirthDay (D, O) – дата рождения;</li> <li>– Person_Sex_id (N, O) – пол;</li> <li>– PersonPhone_Phone (N, H) – телефон;</li> <li>– PersonSnils_Snils (N,H) – СНИЛС;</li> <li>– SocStatus_id (N, O) – социальный статус;</li> <li>– UAddress_id (N, H) – адрес регистрации;</li> <li>– PAddress_id (N, H) – адрес проживания;</li> <li>– BAddress_id (N, H) – адрес рождения;</li> <li>– Org_id (N, H) – место работы;</li> <li>– Post_id (N, H) – должность;</li> <li>– PersonInn_Inn (N, H) – ИНН;</li> <li>– KLCountry_id (N,H) – идентификатор страны гражданства;</li> <li>– Document_IsTwoNation (N,H) – признак "Двойное гражданство"; параметр принимает значения "0" – нет, "1" – да;</li> <li>– Person_pid (N, Y) – идентификатор представителя несовершеннолетнего; если DeputyKind_id не равен "1" ("0", "Отсутствует"), тогда параметр обязателен для заполнения;</li> </ul>
---------------------------	---

	<ul style="list-style-type: none"> <li>- DeputyKind_id (N, Y) – идентификатор типа попечителя; обязателен для заполнения, если указан Person_pid;</li> <li>- Employment_id (N, H) – идентификатор занятости пациента;</li> <li>- FamilyStatus_id (N,H) – идентификатор семейного положения пациента;</li> <li>- PersonFamilyStatus_IsMarried (N,H) – идентификатор типа нахождения в зарегистрированном браке;</li> <li>- DocumentAuthority_id (N,H) – идентификатор типа документа законного представителя;</li> <li>- DocumentDeputy_Ser (T, H) – серия документа законного представителя;</li> <li>- DocumentDeputy_Num (T, H) – номер документа законного представителя;</li> <li>- DocumentDeputy_Issue (T, H) – кем выдан документ законного представителя;</li> <li>- DocumentDeputy_begDate (D, H) – дата выдачи документа законного представителя;</li> <li>- fromMobile (N,H) – признак мобильного телефона;</li> <li>- options (S, H) – настройка работы метода. Допустимые форматы: JSON или массива. Параметры настройки:</li> <li>- CheckChecksumSnilsOff (N, H): <ul style="list-style-type: none"> <li>- 1 – отключить проверку контрольной суммы СНИЛС;</li> <li>- 0 или пропущен – проверять контрольную сумму СНИЛС.</li> </ul> </li> <li>- CheckDuplicateSnilsOff (N,H): <ul style="list-style-type: none"> <li>- 1 – отключить проверку дублирования СНИЛС;</li> <li>- 0 или пропущен – проверять СНИЛС на дублирование</li> </ul> </li> <li>- 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;</li> <li>- OmsSprTerr_id (N,H) – территория страхования;</li> <li>- OrgSMO_id (N,H) – полис выдан;</li> <li>- PolisType_id (N,H) – тип полиса;</li> <li>- PolisFormType_id (N,H) – форма полиса;</li> <li>- Polis_EdNum (T ,H) – ед. номер полиса;</li> <li>- Polis_begDate (DT, N) – дата выдачи полиса;</li> <li>- Polis_endDate (DT, N) – дата закрытия полиса;</li> <li>- Polis_Ser (T ,H) – серия полиса;</li> <li>- Polis_Num (T ,H) – номер полиса</li> </ul>
<p><b>Ответ</b></p>	<p>В случае корректной обработки запроса и успешного создания записи в ответе будет прислано:</p> <ul style="list-style-type: none"> <li>- код статуса: 200</li> <li>- значение параметров: <ul style="list-style-type: none"> <li>- error_code: 0</li> <li>- data: <ul style="list-style-type: none"> <li>- Person_id: значение параметра</li> </ul> </li> </ul> </li> </ul> <p>Если в запросе не заполнен хотя бы один из обязательных параметров, то:</p> <ul style="list-style-type: none"> <li>- код статуса: 200</li> <li>- значение параметров: <ul style="list-style-type: none"> <li>- error_code: 6</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>- error_msg:"Отсутствует один из обязательных параметров PersonSurName_SurName, PersonBirthDay_BirthDay, Person_Sex_id, SocStatus_id"</li> </ul> <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"> <li>- &lt;=7;</li> <li>- &gt;= возраст на текущую дату &lt;= 18;</li> <li>- &gt;18 возраст на текущую дату &lt;=60;</li> <li>- возраст на текущую дату &gt; 60;</li> </ul>
<p><b>Описание ошибок</b></p>	<ul style="list-style-type: none"> <li>- 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";</li> <li>- значение признака "Личность неизвестна" (Person_isUnknown) "Да" и заполнены не все обязательные параметры – отображается ошибка "Отсутствует параметр PersonSurName_SurName";</li> <li>- в БД есть человек с такими же параметрами, которые указаны во входных параметрах метода – отображается ошибка "Человек с такими данными уже присутствует в базе";</li> <li>- параметр PersonPhone_Phone содержит не десятизначное число – отображается ошибка "Неверный формат телефонного номера. Телефонный номер должен состоять из 10 цифр";</li> <li>- в любом передаваемом параметре передается значение, не соответствующее ожидаемому формату – отображается ошибка "Неверный формат поля &lt;Наименования полей через запятую&gt;";</li> <li>- ожидается параметр, состоящий из 11 символов, а передан параметр, состоящий из большего или меньшего количества символов – отображается ошибка "Номер СНИЛС должен содержать 11 символов";</li> <li>- переданная дата рождения больше текущей даты – отображается ошибка "Некорректное значение в поле PersonBirthDay_BirthDay";</li> <li>- 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 &lt;Наименования полей через запятую&gt; не найдены в справочниках Системы";</li> <li>- значение параметров, по которым ожидается значение 0/1 (да/нет) (Person_isUnknown), не найденное в Системе – отображается ошибка "Значение поля &lt;Названия полей через запятую&gt; не соответствует ожидаемому";</li> <li>- в общих настройках параметров Системы в Контроле на</li> </ul>

	<p>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";</p> <ul style="list-style-type: none"> <li>– включена проверка контрольной суммы СНИЛС, и она не пройдена – отображается ошибка "Ошибка проверки контрольной суммы в СНИЛС";</li> <li>– включена проверка СНИЛС на дублирование, и она не пройдена – отображается ошибка "Человек с введённым номером СНИЛС уже есть в базе"</li> </ul>
--	--

#### 7.1.4. Редактирование данных человека PUT api/Person

Метод предназначен для редактирования информации по человеку.

Таблица 11 – Параметры метода PUT api/Person

<b>Входящие параметры</b>	<ul style="list-style-type: none"> <li>– Person_id (N, O) – идентификатор пациента;</li> <li>– PersonSurName_SurName (S[50], H) – фамилия;</li> <li>– PersonFirName_FirName (S[50], H) – имя;</li> <li>– PersonSecName_SecName (S[50], H) – отчество;</li> <li>– PersonBirthDay_BirthDay (D, H) – дата рождения;</li> <li>– Person_Sex_id (N, H) – пол;</li> <li>– PersonPhone_Phone (N[10], H) – телефон;</li> <li>– PersonSnils_Snils (N[11],H) – СНИЛС;</li> <li>– SocStatus_id (N, H) – социальный статус;</li> <li>– UAddress_id (N, H) – адрес регистрации;</li> <li>– PAddress_id (N, H) – адрес проживания;</li> <li>– BAddress_id (N, H) – адрес рождения;</li> <li>– Org_id (N, H) – место работы;</li> <li>– Post_id (N, H) – должность;</li> <li>– PersonInn_Inn (N, H) – ИНН;</li> <li>– KLCountry_id (N,H) – идентификатор страны гражданства;</li> <li>– Document_IsTwoNation (N, H) – признак "Двойное гражданство"; параметр принимает значения "0" – нет, "1" – да;</li> <li>– Person_pid (N, Y) – идентификатор представителя несовершеннолетнего; если DeputyKind_id не равен "1" ("0", "Отсутствует"), тогда параметр обязателен для заполнения;</li> <li>– DeputyKind_id (N, Y) – идентификатор типа попечителя; обязателен для заполнения, если указан Person_pid;</li> <li>– Employment_id (N, H) – идентификатор занятости пациента;</li> <li>– FamilyStatus_id (N,H) – идентификатор семейного положения пациента;</li> <li>– PersonFamilyStatus_IsMarried (N,H) – идентификатор типа нахождения в зарегистрированном браке;</li> <li>– DocumentAuthority_id (N,H) – идентификатор типа документа</li> </ul>
---------------------------	--

	<p>законно представителя;</p> <ul style="list-style-type: none"> <li>– DocumentDeputy_Ser (Т, Н) – серия документа законного представителя;</li> <li>– DocumentDeputy_Num (Т, Н) – номер документа законного представителя;</li> <li>– DocumentDeputy_Issue (Т, Н) – кем выдан документ законного представителя;</li> <li>– DocumentDeputy_begDate (D,H) – дата выдачи документа законного представителя;</li> <li>– DeputyOrg_id (N,H) – идентификатор организации-попечителя;</li> <li>– CheckDeputyTies (N,H) – признак: контролировать допустимость типа представителя (Если контроль включён и тип недопустим, то метод вернёт ошибку);</li> <li>– options (S, H) – настройка работы метода. Допустимые форматы: JSON или массива. Параметры настройки:</li> <li>– CheckChecksumSnilsOff (N, H): <ul style="list-style-type: none"> <li>– 1 – отключить проверку контрольной суммы СНИЛС;</li> <li>– 0 или пропущен – проверять контрольную сумму СНИЛС;</li> </ul> </li> <li>– CheckDuplicateSnilsOff (N,H): <ul style="list-style-type: none"> <li>– 1 – отключить проверку дублирования СНИЛС;</li> <li>– 0 или пропущен – проверять СНИЛС на дублирование;</li> </ul> </li> <li>– DuplicateSearch (Т,H) – список дополнительных параметров для поиска "двойников". Допустимы форматы: JSON, список. Допустимые параметры: PersonSnils_Snils, PersonInn_Inn, Polis_EdNum.</li> </ul> <p>При передаче пустых параметров можно очистить все поля пациента, кроме PersonSurName_SurName, PersonBirthDay_BirthDay, Person_Sex_id, SocStatus_id.</p> <p>Текущие правила:</p> <ul style="list-style-type: none"> <li>– Нельзя указать одновременно Person_pid и DeputyOrg_id. Один из них должен быть пустым или не указан.</li> <li>– Если параметр DeputyKind_id указан и пустой – запись об организации-попечителе удаляется.</li> <li>– Если параметр DeputyKind_id указан и =1 (тип "Отсутствует") – информация об организации-попечителе удаляется. При этом параметр DeputyOrg_id игнорируется.</li> <li>– Если указан только DeputyOrg_id и он пустой – текущая запись об организации-попечителе удаляется.</li> <li>– 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).</li> <li>– Если представитель добавляется (т.е. представителя ещё нет у пациента) – параметр DeputyKind_id становится обязательным</li> </ul>
<b>Ответ</b>	<ul style="list-style-type: none"> <li>– Проводится проверка: параметр PersonPhone_Phone может содержать только 10–значное число. Если результат проверки</li> </ul>

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:
- 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 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): ...

Если параметр не указан, то он не проверяется.

Если параметр указан и пустой, то метод вернёт ошибку: Значения для (PersonSurName\_SurName) обязательное и не может быть пустым.

- PersonBirthDay\_BirthDay (D, У): ...

Если параметр не указан, то он не проверяется.

	<p>Если параметр указан и пустой, то метод вернёт ошибку: Значения для (PersonBirthDay_BirthDay) обязательное и не может быть пустым.</p> <ul style="list-style-type: none"> <li>– Person_Sex_id (N, Y): ...</li> </ul> <p>Если параметр не указан, то он не проверяется.</p> <p>Если параметр указан и пустой, то метод вернёт ошибку: Значения для (Person_Sex_id) обязательное и не может быть пустым..</p> <ul style="list-style-type: none"> <li>– SocStatus_id (N, Y): ...</li> </ul> <p>Если параметр не указан, то он не проверяется.</p> <p>Если параметр указан и пустой, то метод вернёт ошибку: Значения для (SocStatus_id) обязательное и не может быть пустым.</p> <p>Если запрос обработан корректно, и данные успешно обновлены, то:</p> <ul style="list-style-type: none"> <li>– код состояния: 200;</li> <li>– значение параметров:</li> <li>– success: true;</li> <li>– error_code: 0;</li> <li>– data: <ul style="list-style-type: none"> <li>– Person_id: значение параметра.</li> </ul> </li> </ul> <p>Если в запросе не указан обязательный параметр Person_id, то:</p> <ul style="list-style-type: none"> <li>– код состояния: 200;</li> <li>– значение параметров:</li> <li>– error_code: 6;</li> <li>– error_msg: Отсутствует обязательный параметр 'Person_id'.</li> </ul> <p>Если в запросе указан неверный формат номера:</p> <ul style="list-style-type: none"> <li>– код состояния: 200;</li> <li>– значение параметров:</li> <li>– success: false;</li> <li>– error_code: 6;</li> <li>– error_msg: Неверный формат телефонного номера. Телефонный номер должен состоять из 10 цифр</li> </ul>
--	---

**7.1.5. Получение записей на прием к врачу GET  
api/TimeTableGraf/TimeTableGrafByMedStaffFact**

Таблица 12 – Параметры метода **GET api/TimeTableGraf/TimeTableGrafByMedStaffFact**

<b>Входящие параметры</b>	<ul style="list-style-type: none"> <li>– MedStaffFact_id (N, O) – место работы врача</li> <li>– TimeTableGraf_beg (DT, O) – дата и время начала диапазона</li> <li>– TimeTableGraf_end (DT, O) – дата и время окончания диапазона</li> </ul>
---------------------------	--

<b>Ответ</b>	<p>Успешный ответ – список бирок и пациентов, записанных к врачу в указанный диапазон:</p> <ul style="list-style-type: none"> <li>– TimeTableGraf_id (N, O) – идентификатор бирки</li> <li>– TimeTableGraf_begTime (DT, O) – дата и время приема</li> <li>– TimeTableGraf_factTime (DT, H) – фактическое время приема.</li> <li>– Person_id (N, O) – идентификатор пациента</li> </ul>
--------------	--

### 7.1.6. Получение прикреплений пациента GET api/PersonAttach

Таблица 13 – Параметры метода GET api/PersonAttach

<b>Входящие параметры</b>	<ul style="list-style-type: none"> <li>– Person_id (N, O) – идентификатор пациента;</li> <li>– Lpu_id (N, H) – идентификатор МО;</li> <li>– LpuRegion_id (N, H) – идентификатор участка;</li> <li>– LpuAttachType_id (N, H) – идентификатор типа прикрепления</li> </ul>
<b>Ответ</b>	<p>Ошибки:</p> <p>Если person_id не найден, то метод возвращает "error_code", равный 0.</p> <p>Если не заполнен обязательный параметр person_id, то метод возвращает "error_code", равный 6, и сообщение об ошибке: "Отсутствует обязательный параметр 'Person_id'".</p> <p>Успешный ответ:</p> <ul style="list-style-type: none"> <li>– PersonSnils_Snils (N[11], H) – СНИЛС;</li> <li>– Org_id (N, H) – место работы;</li> <li>– Post_id (N, H) – должность.</li> </ul> <p>Список прикреплений:</p> <ul style="list-style-type: none"> <li>– PersonCard_id (N, H) – идентификатор прикрепления;</li> <li>– PersonCard_begDate (D, H) – дата прикрепления;</li> <li>– PersonCard_endDate (D, H) – дата открепления;</li> <li>– PersonCard_Code (N, H) – номер амбулаторной карты;</li> <li>– Lpu_id – идентификатор МО прикрепления;</li> <li>– Lpu_Name – полное наименование МО прикрепления;</li> <li>– Lpu_Nick – краткое наименование МО прикрепления;</li> <li>– LpuRegion_Name – наименование участка;</li> <li>– LpuRegion_id (N, H) – идентификатор участка прикрепления;</li> <li>– LpuAttachType_id (N, H) – идентификатор типа прикрепления;</li> <li>– PersonCard_isAttachAuto (N, H) – признак автоприкрепления;</li> <li>– CardCloseCause_id (N, H) – идентификатор причины закрытия карты;</li> <li>– PersonCard_AttachAutoDT (DT, H) – дата и время автоматического прикрепления;</li> <li>– PersonCard_isAttachCondit (N, H) – признак условного прикрепления</li> </ul>

<b>Пример</b>	<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 №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"         }       ]     }   ] } </pre>
---------------	---

### 7.1.7. Получение основного прикрепления пациента GET api/PersonMainAttach

Таблица 14 – Параметры метода GET api/PersonMainAttach

<b>Назначение метода</b>	Возвращаются данные по МО действующего основного прикрепления пациента (тип участка – терапевтический или педиатрический, если пациенту меньше 18 лет), а также данные участкового врача.
<b>Входящие параметры</b>	Person_id (N, O) – идентификатор пациента.
<b>Ответ</b>	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 определения 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, N) – 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

<p><b>Ошибки</b></p>	<p>Возможные ошибки:</p> <ul style="list-style-type: none"> <li>- если в запросе не заполнен входящий параметр, отобразится ошибка (прикладной код: "6") с текстом: "Отсутствует обязательный параметр 'Person_id'";</li> <li>- если по Person_id не найдено действующего основного прикрепления пациента, отобразится ошибка (прикладной код: "1") с текстом: "Не найдено действующего основного прикрепления пациента!"</li> </ul>
<p><b>Пример</b></p>	<p>Успешный ответ:</p> <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\", "Address_Address": "614068, РОССИЯ, ПЕРМСКИЙ КРАЙ, Г ПЕРМЬ, ЛУНАЧАРСКОГО УЛ, д. 90, ", "Org_phone": "2657273", "MedStaffFact_id": null, "Person_Fio": null } } </pre>

## 5.5 Общие методы

### 5.5.1 Получение элементов справочника GET api/Refbook

Метод используется для получения элементов справочника.

Таблица 15 – Параметры метода GET api/Refbook

<b>Входящие параметры</b>	<ul style="list-style-type: none"><li>– Refbook_Code – Код справочника;</li><li>– Refbook_TableName – Наименование таблицы справочника;</li><li>– Code – Код элемента справочника;</li><li>– Id – идентификатор элемента справочника;</li><li>– Name – Наименование элемента справочника</li><li>– Offset – Номер страницы;</li><li>– Limit – Максимальное число возвращаемых записей;</li><li>– Page – номер начальной возвращаемой записи</li></ul>
<b>Пример запроса</b>	GET {{url}}/api/Refbook?Refbook_TableName=dbo.socstatus&Page=2&limit=2
<b>Ответ</b>	<ul style="list-style-type: none"><li>– data:<ul style="list-style-type: none"><li>– id – Идентификатор элемента справочника;</li><li>– OrgType_id – Идентификатор типа организации;</li><li>– Name – Наименование элемента справочника;</li><li>– Code – Код элемента справочника;</li><li>– KLRgn_id – Идентификатор региона;</li><li>– begDate – Дата начала;</li><li>– endDate – Дата окончания</li></ul></li></ul>
<b>Пример ответа</b>	<pre>{   "error_code": 0,   "data": [     {       "id": "10000223",       "OrgType_id": null,       "Name": "БОМЖ",       "Code": "3",       "KLRgn_id": null,       "begDate": null,       "endDate": "2020-05-01"     },     {       "id": "10000224",       "OrgType_id": null,</pre>

	<pre> &gt;Name": "Военнослужащий МВД", "Code": "4", "KLRgn_id": null, "begDate": null, "endDate": "2020-05-01"     }   ] } </pre>
<p><b>Ошибки</b></p>	<ul style="list-style-type: none"> <li>- Проверить наличие обязательных параметров: <ul style="list-style-type: none"> <li>- Если в запросе должен быть хотя бы один из перечисленных обязательных параметров, то сформировать ошибку 6 Отсутствует хотя бы один из параметров [Refbook_Code, Refbook_TableName]. Перейти к шагу 4.</li> <li>- Иначе перейти к следующему шагу.</li> </ul> </li> <li>- Провалидировать значения входящих параметров. Если нарушены правила валидации, то сформировать ошибку из списка ниже. <ul style="list-style-type: none"> <li>- Если значение параметра, являющегося идентификатором, не является целым числом больше нуля, то сформировать ошибку 6 Ошибка формата идентификатора. Перейти к шагу 4.</li> <li>- Если переданные значения не соответствуют заявленным типам, то сформировать ошибку 6 Ошибка соответствия параметров заявленным типам. Перейти к шагу 4.</li> </ul> </li> <li>- Проверить данные справочников: <ul style="list-style-type: none"> <li>- Если данные справочника (Refbook_Code, Refbook_TableName) переданные во входных параметрах отсутствуют в списке внутренних справочников РИШ (dbo.RISHRefbook), то сформировать ошибку 6 Данного справочника нет в списке Внутренних справочников РИШ. Перейти к шагу 4.</li> <li>- Если найдено больше одного справочника подходящего под условия запроса, то сформировать ошибку 6 По указанному коду найдено несколько справочников. Перейти к шагу 4.</li> <li>- Если поле отсутствует в запрашиваемом справочнике, то сформировать ошибку 6 Поля нет в справочнике. Перейти к шагу 4.</li> </ul> </li> </ul>

Алгоритм завершает работу
---------------------------

### 5.5.2 Поиск человека по атрибутам GET /api/Person/mgetPersonSearch

Мобильный метод предназначен для поиска человека по атрибутам.

Таблица 16 – Параметры метода GET /api/Person/mgetPersonSearch

<b>Входящие параметры</b>	<p>Условия формирования параметров запроса:</p> <ul style="list-style-type: none"><li>– Если передается единственный параметр - номер КВС, то он обязательный;</li><li>– Если передается Идентификатор МО, то обязательно передавать еще номер КВС;</li><li>– Если Идентификатор МО не передается, то номер КВС не обязательно передавать;</li><li>– Если передается другой параметр запроса (не Идентификатор МО и не номер КВС), тогда надо обязательно указать два параметра запроса;</li><li>– Возможные значения в параметре запроса searchMode (по умолчанию=all):<ul style="list-style-type: none"><li>– wow - цепляем участников ВОВ;</li><li>– attachrecipients - только по льготникам и прикрепленным;</li><li>– evnpldispscreen - если ищем для скринингового исследования, то возраст пациента на конец выбранного года должен соответствовать</li><li>– evnpldtipre - если ищем по картам первого этапа предвостмотра несовершеннолетних</li><li>– evnpldtipro - если ищем по картам первого этапа профосмотра несовершеннолетних</li><li>– withlgotonly - только по льготникам</li><li>– dddispclass2 - ДВН 2</li><li>– ddorppperiod - периодический осмотр</li><li>– ddorppsec - если ищем по картам первого этапа детей-сирот</li><li>– hasorenevnpns - если ищем по открытым КВС</li><li>– ddorp или ddorp_all - сли ищем по регистру детей-сирот</li><li>– geriatics - старше 60 лет</li><li>– dt14 - если ищем по ДД 14</li><li>– older14notdead - только старше 14 лет и не умершие</li><li>– dt6 - дети младше 6 лет</li><li>– palliat или narc - хз, но вроде ищем без даты смерти</li></ul></li></ul>
---------------------------	--

	<ul style="list-style-type: none"> <li>- attbefore3 - только прикрепленные или старше 3 лет</li> <li>- att_vol - только прикрепленные и в объёме</li> <li>- att_1_4 - только прикрепленные (основное или служебное)</li> <li>- att - только прикрепленные</li> <li>- person_register_by_vzn_request - поиск в регистре для заявки ВЗН</li> <li>- hms - поиск для регистра главных внештатных специалистов</li> <li>- fmba - только не включенные в регистр ФМБА</li> <li>- dd - если ищем по ДД</li> <li>- encryonly - отображение только зашифрованных ВИЧ-инфицированных</li> <li>- men_only - women_only - только эм или жо</li> <li>- smo или smo3 - Поиск в арм смо. Для Перми - последние 3 года</li> <li>- erssnils - снилс должен быть заполнен</li> </ul>
<p><b>Пример</b></p>	<p><b>Успешный ответ:</b></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", </pre>

```
"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,
"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,

	<pre> "accessType": "edit"     }   ] } </pre>
<b>Ответ</b>	<p>Логика формирования параметров ответа:</p> <p>Person_IsBDZ. Значение по умолчанию:</p> <ul style="list-style-type: none"> <li>- Если Polis_endDate не пустое поле и Polis_endDate меньше текущей даты, то 'orange'.</li> <li>- Если person.PersonCloseCause_id= 2 и person.Person_closeDT не пустое поле, то 'red'</li> <li>- Если person.server_id = 0, то 'true'Иначе 'false'</li> <li>- Для Перми: <ul style="list-style-type: none"> <li>- Если person.server_id = 0, то 'true'</li> <li>- Если person.Person_IsInErz = 1, то 'blue'</li> <li>- Если person.server_id = 0 и Polis_endDate меньше текущей даты, то</li> <li>- Если Person_deadDT - пустое поле, то 'yellow'</li> <li>Иначе - 'red'</li> <li>- Если person.server_id = 2 и person.Person_IsInErz не равно 1, то 'false'</li> </ul> </li> <li>- Для Пензы: <ul style="list-style-type: none"> <li>- Если person.Person_IsInErz = 1, то 'orange'</li> <li>- Если person.Person_IsInErz = 2, то 'true'</li> <li>- Иначе 'false'</li> </ul> </li> <li>- Для Казахстана - <ul style="list-style-type: none"> <li>- Если person.Person_IsInErz = 1, то 'red'</li> <li>- Если person.Person_IsInErz = 2, то 'true'Иначе 'false'</li> </ul> </li> </ul>
<b>Ошибки</b>	<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>

### 5.5.3 Работа с направлениями

#### 5.5.3.1 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\_stdid и не передан входящий параметр Lpu\_sid, то заполняется значением 2;
      - Org\_sid – направившая организация. Если передан параметр PassportToken\_stdid 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\_stdid. Если организация не найдена,

- отображается ошибка с текстом: "Не найдена организация по параметру PassportToken\_stid", 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\_stid 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\_stid, 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".

Таблица 17 – Параметры метода **POST api/EvnDirection**

<b>Входящие параметры</b>	<ul style="list-style-type: none"> <li>– 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;</li> <li>– Person_id (N, H) – идентификатор пациента;</li> <li>– EvnDirection_Num (T, H) – номер направления;</li> </ul>
---------------------------	---

- 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;
- 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

	<p>отделения;</p> <ul style="list-style-type: none"> <li>– Lpu_did (N, O) – идентификатор МО, куда направили;</li> <li>– LpuSection_did (N, H) – идентификатор отделения, куда направили;</li> <li>– 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&lt;&gt;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";</li> <li>– MedPersonal_did (N, H) – идентификатор врача, к кому направили;</li> <li>– LpuUnit_did (N, H) – идентификатор группы отделений, куда направили;</li> <li>– 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&lt;&gt;2). Если параметр не передан, отображается ошибка: "Отсутствуют обязательные параметры для направления на телемедицинскую консультацию: TimetableGraf_id";</li> <li>– TimetableStac_id (N, H) – идентификатор стационарной бирки;</li> <li>– 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";</li> <li>– TimeTableResource_id (N, H) – идентификатор диагностической бирки;</li> <li>– EvnPrescr_id (N, H) – идентификатор назначения;</li> </ul>
--	---

- 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

	<p>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";</p> <ul style="list-style-type: none"> <li>- LpuUnitType_id (N, H) – идентификатор условия оказания медицинской помощи;</li> <li>- UslugaComplex_id (N, H) – идентификатор услуги;</li> <li>- 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);</li> <li>- 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&lt;&gt;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);</li> <li>- 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&lt;&gt;2) или "Телемедицинский консилиум" (если DirType_id = 17 и EvnDirection_IsConcilium=2);</li> <li>- MedPost_Code (T, Y) – код должности направившего врача из справочника "Должности работников организаций медицинского и</li> </ul>
--	--

	<p>фармацевтического профиля 1.2.643.5.1.13.13.11.1002".</p> <p>Обязательный параметр для заполнения, если:</p> <ul style="list-style-type: none"> <li>– не передан входящий параметр MedStaffFact_id;</li> <li>– для направлений на удаленную консультацию в формате "Врач-врач" (если DirType_id = 17 и RequestConsult_id=2 и EvnDirection_IsConcilium&lt;&gt;2) или "Телемедицинский консилиум" (если DirType_id = 17 и EvnDirection_IsConcilium=2)";</li> <li>– MedPersonal_ExternalId (Т, Н) – идентификатор врача из внешней системы;</li> <li>– 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";</li> <li>– 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";</li> <li>– 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";</li> <li>– DiagSetPhase_id (N, У) – идентификатор общего состояния пациента</li> </ul>
--	---

	<ul style="list-style-type: none"> <li>– TestList(A, H) – список тестов исследования; если массив не заполнен, то в направлении должны быть назначены все тесты исследования:</li> <li>– UslugaComplexMedService_id (N, O) – идентификатор теста.</li> <li>– EvnDirectionOper_IsAgree (N, H) – идентификатор согласия на операцию.</li> </ul> <p>Параметры для направления на прижизненное патологоанатомическое исследование биопсийного (операционного) материала. Данные поля должны быть заполнены в случае, если DirType_id = 7</p>
<b>Выходные параметры</b>	<p>Если запрос обработан успешно, и в БД создана запись, то в ответе будет отправлено:</p> <ul style="list-style-type: none"> <li>– EvnDirection_id (N, H) – идентификатор направления на лабораторное исследование;</li> <li>– EvnConcilium_id (N, H) – идентификатор консилиума;</li> <li>– Evn_id (N, H) – идентификатор события;</li> <li>– EvnQueue_id (N, H) – идентификатор постановки в очередь;</li> <li>– EvnLabRequest_id (N, H) – идентификатор заявки на лабораторное исследование;</li> <li>– 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;</li> <li>– EvnDirectionHistologic_id (N, H) – идентификатор в таблице направления на прижизненное патологоанатомическое исследование биопсийного (операционного) материала</li> </ul>
<b>Пример</b>	<pre>{   "error_code": 0,   "data": {     "EvnDirection_id": "140101014227728",     "Evn_id": "140101014227728",     "EvnLabRequest_id": "140101014227729",     "EvnQueue_id": "140101014227727",     "EvnPrescr_id": "140101014227726"   } }</pre>

## 5.6 Методы работы с листами ожидания

### 5.6.1 Добавление записи в лист ожидания POST api/EvnQueue/EvnQueue

Таблица 18 – Параметры метода POST api/EvnQueue/EvnQueue

<p><b>Входящие параметры</b></p>	<ul style="list-style-type: none"> <li>– Person_id (N, O) – идентификатор пациента;</li> <li>– LpuSectionProfile_id (N, Y) – идентификатор профиля отделения МО.</li> <li>– LpuSection_id (N, H) – идентификатор отделения.</li> <li>– MedPersonal_id (N, H) – направивший врач;</li> <li>– MedStaffFact_did (N, Y) – врач, к кому в лист ожидания включается пациент. Указывается, если запись выполняется к конкретному врачу.</li> <li>– Lpu_sid (N, H) – направившее МО;</li> <li>– Lpu_did (N, O) – МО, куда направили;</li> <li>– EvnQueue_desDT (D, H) – дата и время посещения (желаемая);</li> <li>– MedService_did (N, Y) – идентификатор службы, в которую нужно поставить в очередь;</li> <li>– UslugaComplex_did (N, O) – идентификатор услуги, на которую необходимо записать</li> </ul>
<p><b>Действия при получении запроса</b></p>	<p>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.</p> <p>б) Если по переданному значению Person_id не найден пациент, то ошибка "Пациент не найден в системе".</p> <p>в) Если запись с аналогичными параметрами найдена в системе, то ошибка "Данные записи не прошли проверку на дублирование". Создание записи не производится.</p> <p>г) Если в любом передаваемом параметре передается значение, не соответствующее ожидаемому формату, например:</p> <ul style="list-style-type: none"> <li>– ожидается числовое значение, а передано текстовое значение;</li> <li>– ожидается параметр, состоящий из 11 символов, а передан параметр, состоящий из большего или меньшего количества символов;</li> </ul> <p>то, ошибка: "Неверный формат поля &lt;Наименования полей через запятую&gt;", создание записи в БД не происходит.</p> <p>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 &lt;Наименования полей через запятую&gt; не найдены в справочниках системы". Создание записи в БД не происходит.</p>

	<p>Если есть ошибки, то возвращается ошибка. Иначе:</p> <ul style="list-style-type: none"> <li>– в лист ожидания добавляется запись – пациент ставится в очередь по профилю или врачу (в соответствии с полученными параметрами).</li> <li>– статус записи "в очереди" (EvnQueueStatus_id).</li> <li>– способ записи/включения в Лист ожидания – "РИШ"</li> </ul>
<b>Ответ</b>	<p>Если есть ошибки, то возвращается код ошибки. Иначе:</p> <ul style="list-style-type: none"> <li>– EvnQueue_id (N, O) – идентификатор постановки в очередь</li> <li>– EvnQueueStatus_id (N, H) – статус листа ожидания</li> </ul>

## 5.6.2 Получение листа ожидания GET api/EvnQueue/EvnQueue

Таблица 19 – Параметры метода GET api/EvnQueue/EvnQueue

<b>Входящие параметры</b>	<ul style="list-style-type: none"> <li>– Person_id (N, Y) – идентификатор пациента. Обязательно указывается, если не заполнены другие условные параметры;</li> <li>– Lpu_id (N, Y) – идентификатор МО в справочнике РИШ. Обязательно указывается, если не указан пациент;</li> <li>– LpuSection_id (N, Y) – идентификатор отделения.</li> <li>– LpuSectionProfile_id (N, Y) – идентификатор профиля отделения МО. Обязательно указывается, если не указан пациент и отделение и врач.</li> <li>– MedStaffFact_id (N, Y) – врач, к кому в лист ожидания включается пациент.)</li> </ul>
<b>Действия</b>	<p>Выполняется проверка заполнения обязательных параметров, в т.ч.</p> <ul style="list-style-type: none"> <li>– если не заполнен ни один из условных параметров или обязательный параметр, то ошибка.</li> </ul> <p>Если есть ошибки то, возвращается код ошибки. Иначе: Возвращаются записи очереди, удовлетворяющие заданным входящим параметрам</p> <ul style="list-style-type: none"> <li>– имеют статус "в очереди";</li> <li>– или у которых направления имеют статус "Поставлено в очередь".</li> </ul>
<b>Ответ</b>	<p>Если есть ошибки, то возвращается код ошибки. Иначе: выдаются следующие данные:</p> <ul style="list-style-type: none"> <li>– EvnQueue_id (N, O) – идентификатор постановки в очередь</li> <li>– Person_id (N) – Идентификатор пациента.</li> <li>– Lpu_id (N) – идентификатор МО очереди в справочнике РИШ.</li> </ul>

	<ul style="list-style-type: none"> <li>– LpuSectionProfile_id (N) – идентификатор профиля отделения МО.</li> <li>– LpuSection_id (N) – идентификатор отделения</li> <li>– MedStaffFact_id (N) – врач, к кому в лист ожидания включен пациент (значение сущности persis.MedWorker);</li> <li>– EvnQueue_setDate (D) – дата постановки в очередь;</li> <li>– EvnQueue_setTime (T) – время постановки в очередь</li> <li>– EvnQueueStatus_id (N) – статус записи в очереди.</li> <li>– EvnQueue_desDT (D, H) – желаемые дата и время посещения</li> </ul>
--	--

## 5.7 Методы сервиса "Запись к врачу"

### 5.7.1 Получение должности по наименованию GET api/PostByName

Метод предназначен для получения должности по наименованию.

Таблица 20 – Параметры метода GET api/PostByName

<b>Входящие параметры</b>	<ul style="list-style-type: none"> <li>– post_name (T, O) – наименование должности;</li> <li>– sess_id (N, O) – идентификатор сессии</li> </ul>
<b>Ответ</b>	<p>Если запрос был обработан успешно и в БД была найдена хотя бы одна запись, то в ответе будет отправлено:</p> <ul style="list-style-type: none"> <li>– error_code: 0</li> <li>– data:</li> <li>– Post_id:</li> <li>– Post_name:</li> </ul> <p>В ответе метода должны быть отправлены все записи, по которым было установлено соответствие со значением должности из запроса</p>

### 5.7.2 Получение МО по идентификатору GET api/МОById

Таблица 21 – Параметры метода GET api/МОById

<b>Входящие параметры</b>	<ul style="list-style-type: none"> <li>– Lpu_id (N, O) – Идентификатор МО</li> </ul>
<b>Ответ</b>	<p>Успешный ответ</p> <ul style="list-style-type: none"> <li>– Org_Name (T, O) – Наименование медицинской организации</li> <li>– Org_Nick (T, O) – Краткое наименование медицинской организации</li> </ul>

### 5.7.3 Получение информации о докторе GET /api/doctors/MedStaffFact\_id

Метод позволяет получить информацию о враче.

Таблица 22 – Параметры метода **GET /api/doctors/MedStaffFact\_id**

<p><b>Входящие параметры</b></p>	<ul style="list-style-type: none"> <li>– version (T, O) – версия;</li> <li>– sess_id (T, O) – идентификатор сессии;</li> <li>– apiKey (T, O) – ключ апи;</li> <li>– MedStaffFact_id (T, O) – идентификатор врача;</li> <li>– answerAsInGetMethod (N, H) – фильтр</li> </ul>
<p><b>Выходные параметры</b></p>	<ul style="list-style-type: none"> <li>– id (T) – идентификатор врача;</li> <li>– doctor_fio (T) – Ф. И. О врач;</li> <li>– profile_name (T) – название профиля врача;</li> <li>– profile_id (T) – идентификатор профиля;</li> <li>– category (T) – врачебная категория;</li> <li>– lpu_nick (T) – краткое название больниц;</li> <li>– unit_name (T) – название подразделения;</li> <li>– address (T) – адрес подразделения;</li> <li>– feedbacks (T) – отзывы о враче:</li> <li>– count (N) – количество отзывов;</li> <li>– comments_count (N) – количество комментариев;</li> <li>– rating_avg (N) – средняя оценка;</li> <li>– Lpu_id (T) – идентификатор больницы;</li> <li>– current_main (S) – основное место работы врача;</li> <li>– FullName (T) – Ф. И. О. врача;</li> <li>– QualificationCat_Name (T) – квалификационная категория;</li> <li>– QualificationCategory (T) – степень квалификационной категории;</li> <li>– Person_id (T) – идентификатор врача;</li> <li>– WorkData_begDate (S) – дата начала работы;</li> <li>– WorkType_id (T) – идентификатор типа занимаемой должности;</li> <li>– Dolgnost_Name (T) – название должности;</li> <li>– Lpu_id (T) – идентификатор больницы;</li> <li>– Lpu_Name (T) – название больниц;</li> <li>– MedSpecOms_Name (T) – специальность врача;</li> <li>– MedSpecOms_Code (T) – код специальности;</li> <li>– LpuSection_Name (T) – название отделения;</li> <li>– LpuUnit_Name (T) – название подразделения;</li> <li>– Address_id (T) – идентификатор адреса;</li> <li>– MedStaffFact_id (T) – идентификатор рабочего места врача;</li> <li>– MedPersonal_id (T) – идентификатор медработника;</li> <li>– Age (N) – возраст;</li> <li>– LpuUnit_Address (T) – адрес подразделения;</li> <li>– current_addons (S) – дополнительное место работы врача</li> <li>– dolgnost (T) – название должности;</li> <li>– educations (S) – данные об образовании:</li> <li>– YearOfGraduation (N) – год окончания учёбы;</li> <li>– EducationType_Name (T) – тип обучения;</li> <li>– EducationInstitution_Name (T) – название учебного заведения;</li> <li>– Speciality_id (T) – идентификатор специальности;</li> <li>– DiplomaSpeciality_Name (T) – специальность в дипломе;</li> <li>– regions (T) – регион;</li> <li>– region_streets (T) – обслуживаемые улицы;</li> </ul>

	<ul style="list-style-type: none"> <li>- annot (T) – примечание;</li> <li>- status (N) – статус. Возможные значения:</li> <li>- 1 – есть ближайшая свободная бирка;</li> <li>- 2 – нет расписания;</li> <li>- status_msg (T) – текст с первой ближайшей записью;</li> <li>- can_post_feedback (B) – возможность комментирования. Возможные значения:</li> <li>- true – да,</li> <li>- false – нет</li> <li>- doctorInfoList (S, H) – информация о врачах при установленном фильтре doctorInfoList:</li> <li>- MedStaffFact_id (S) – идентификатор места работы врача;</li> <li>- IsPaidRec (N) – платный врач;</li> <li>- MedPersonal_FullName (S) – Ф. И. О. врача;</li> <li>- Lpu_Nick (S) – название МО;</li> <li>- LpuUnit_Address (S) – адрес МО;</li> <li>- QualificationCat_Name (S) – квалификация врача;</li> <li>- feedback_total (N) – количество оценок;</li> <li>- feedback_rating (N) – рейтинг врача;</li> <li>- firstFreeDate (S) – дата ближайшей бирки;</li> <li>- ProfileSpec_Name (S) – специальность;</li> <li>- videochatIsOn (B) – флаг - видеочат включен;</li> <li>- IsAttachedDoctor (B) – находится ли доктор в прикреплении у пациента;</li> <li>- rating_avg (N) – рейтинг врача;</li> <li>- avatar_path (S) – аватар врача;</li> <li>- lng (N) – координаты долготы;</li> <li>- lat (N) – координаты широты;</li> <li>- firstFreeDates (S) – дата/время бирки;</li> <li>- source_system (S) – источник поиска врачей: pm_main – основные, pm_paid – платные</li> </ul>
<b>Ответ</b>	<p><b>Успешный ответ:</b></p> <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,</pre>

```

"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"
  },
  "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",

```

	<pre> "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>
--	---

#### 5.7.4 Получение списка специальностей в МО GET api/MedSpecOms/MedSpecOmsByMO

Таблица 23 – Параметры метода GET api/MedSpecOms/MedSpecOmsByMO

<b>Входящие параметры</b>	<ul style="list-style-type: none"> <li>– Lru_id (N, O) – Идентификатор МО</li> <li>– 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.</li> </ul>
<b>Ответ</b>	<p>Успешный ответ – список специальностей в МО, для которых есть хотя бы одно место работы на текущую дату (MedStaffFact).</p> <ul style="list-style-type: none"> <li>– MedSpecOms_id (N, O) – Идентификатор специальности.</li> <li>– TimetableGraf_Count (N, Y) – Количество доступных бирок для записи (выгружается, если For_Record = 1)</li> </ul>

#### 5.7.5 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 постраничного 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.

Таблица 24 – Параметры метода **GET api/MedStaffFact/MedStaffFactByMO**

<p><b>Входящие параметры</b></p>	<ul style="list-style-type: none"> <li>– MedSpecOms_id (N, O) – идентификатор специальности;</li> <li>– Lpu_id (N, O) – идентификатор МО;</li> <li>– Post_id (N, O) – идентификатор должности;</li> <li>– MedStaffFact_IsPaidRec (N, H) – признак наличия платного приёма;</li> <li>– TimeTableType_id (N, H) – тип бирки;</li> <li>– offset (N, H) – номер начальной возвращаемой записи;</li> <li>– limit (N, H) – максимальное число возвращаемых записей;</li> <li>– page (N, H) – номер страницы</li> </ul>
<p><b>Ответ</b></p>	<p>Успешный ответ – массив объектов:</p> <ul style="list-style-type: none"> <li>– MedStaffFact_id (N, O) – место работы врача;</li> <li>– Person_id (N, O) – идентификатор человека по месту работы;</li> <li>– PersonSurName_SurName (S[50], H) – фамилия;</li> <li>– PersonFirName_FirName (S[50], H) – имя;</li> <li>– PersonSecName_SecName (S[50], H) – отчество.</li> <li>– RecType_id (N, H) – тип записи;</li> <li>– LpuUnitType_id (N, O) – тип подразделения МО;</li> <li>– MedStaffFact_setDate (D, O) – дата начала работы сотрудника;</li> <li>– MedStaffFact_disDate (D, H) – дата окончания работы сотрудника;</li> <li>– LpuBuilding_id (N, O) – идентификатор;</li> <li>– Post_id (N, H) – идентификатор должности;</li> <li>– Lpu_id (N, H) – идентификатор МО;</li> <li>– LpuSection_id (N, H) – идентификатор отделения;</li> <li>– 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</li> </ul>

<p><b>Пример</b></p>	<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": "10101000000194"     },     {       "MedStaffFact_id": "140101000000014",       "Person_id": "101010000000060",       "PersonSurName_SurName": "БАНАЛЬНЫЙ",       "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": "140101000001993",       "Person_id": "101010000000002",       "PersonSurName_SurName": "ВРАЧ",       "PersonFirName_FirName": "ТЕСТ",       "PersonSecName_SecName": "",       "RecType_id": null,       "LpuUnitType_id": "9", </pre>
----------------------	---

	<pre> "MedStaffFact_setDate": "01.01.2021", "MedStaffFact_disDate": null, "LpuBuilding_id": "10101000000074"     }   ] }</pre>
--	--

### 5.7.6 Получение свободных дат приема GET api/TimeTableGraf/TimeTableGrafFreeDate

Таблица 25 – Параметры метода GET api/TimeTableGraf/TimeTableGrafFreeDate

<b>Входящие параметры</b>	<ul style="list-style-type: none"> <li>– MedStaffFact_id (N, O) – место работы врача</li> <li>– TimeTableGraf_beg (D, O) – дата начала диапазона</li> <li>– TimeTableGraf_end (D, O) – дата окончания диапазона</li> <li>– TimeTableGrafFreeDate (N, H) – параметр учета прошедшего времени (1 – да, 0 – нет);</li> <li>– TimeTableType_id (N, H) – тип бирки: <ul style="list-style-type: none"> <li>– если указано значение входного параметра – ответ со списком свободных дат, на которых есть бирки с указанным типом;</li> <li>– если пустое значение или не указан – метод работает по старым правилам</li> </ul> </li> </ul>
<b>Ответ</b>	<p>Успешный ответ – список свободных дат</p> <ul style="list-style-type: none"> <li>– TimeTableGraf_begTime (D, O) – даты, где есть бирки со статусом "свободная". Если TimeTableGrafFreeDate=1 и свободные бирки на текущую дату отсутствуют, дата не возвращается.</li> </ul>

### 5.7.7 Получение свободного времени приема GET api/TimeTableGraf/TimeTableGrafFreeTime

Таблица 26 – Параметры метода GET api/TimeTableGraf/TimeTableGrafFreeTime

<b>Входящие параметры</b>	<ul style="list-style-type: none"> <li>– MedStaffFact_id (N, O) – место работы врача;</li> <li>– TimeTableGraf_begTime (D, O) – свободная дата приема;</li> <li>– Person_id (N, H) – идентификатор пациента. Правила: <ul style="list-style-type: none"> <li>– если Person_id не указан, то возвращаются все виды бирок, кроме вида "Диспансерный учёт";</li> <li>– если Person_id указан, и у пациента нет открытой карты диспансерного наблюдения, то возвращаются все виды бирок, кроме вида "Диспансерный учёт";</li> <li>– если Person_id указан и у пациента есть открытая карта диспансерного наблюдения, то возвращаются все виды бирок, в том числе вида "Диспансерный учёт";</li> </ul> </li> </ul>
---------------------------	---

	<ul style="list-style-type: none"> <li>– TimeTableType_id (N, H) – тип бирки;</li> <li>– 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;</li> <li>– offset (N, H) – номер начальной возвращаемой записи;</li> <li>– limit (N, H) – максимальное число возвращаемых записей. Значение по умолчанию – 10000;</li> <li>– page (N, H) – номер страницы. Значение по умолчанию – 1</li> </ul>
<b>Ответ</b>	<p>Успешный ответ – массив свободных бирок на прием к врачу: {</p> <ul style="list-style-type: none"> <li>– 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";</li> <li>– TimeTableGraf_begTime (DT, O) – дата и время начала приема;</li> <li>– TimeTableGraf_Time (N,O) – длительность приема.</li> </ul> <p>}</p> <p>Выходной параметр для случаев, когда бирка, на которую пытаются записаться, заблокирована:</p> <ul style="list-style-type: none"> <li>– TimeTableStatusLockComment (T, H) – сообщение о блокировке бирки. Сообщение содержит в себе возможное время разблокировки бирки.</li> </ul> <p>Проверка на наличие переданной во входных параметрах бирки в таблице БД по TimeTableMedService_id:</p> <ul style="list-style-type: none"> <li>– 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;</li> <li>– в связи с данной проверкой для бирок, которые нашлись в таблице БД выводится в ответе параметр:</li> <li>– 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</li> </ul>

	нее 2017-12-23 21:59:34.260000" (т.е. в конце комментария выводится время из TimeTableLock_lockTime + 15 мин)
--	---

### 5.7.8 Запись пациента на прием POST api/TimeTableGraf/TimeTableGrafWrite

Таблица 27 – Параметры метода POST api/TimeTableGraf/TimeTableGrafWrite

<b>Входящие параметры</b>	<ul style="list-style-type: none"> <li>– Person_id (N, O) – идентификатор пациента;</li> <li>– TimeTableGraf_id (N, O) – идентификатор свободной бирки;</li> <li>– EvnQueue_id (N, H) – идентификатор постановки в очередь</li> </ul>
<b>Ответ</b>	<ul style="list-style-type: none"> <li>– если по переданному значению Person_id не найден пациент, то ошибка "Пациент не найден в системе".</li> <li>– если по переданному значению TimeTableGraf_id не найдена свободная бирка, то ошибка "Не найдена свободная бирка".</li> <li>– если по переданному значению EvnQueue_id не найдена запись в БД, то ошибка "Не найдена запись о постановке в очередь".</li> <li>– если в любом передаваемом параметре передается значение, не соответствующее ожидаемому формату, например: <ul style="list-style-type: none"> <li>– ожидается числовое значение, а передано текстовое значение;</li> <li>– ожидается параметр, состоящий из 11 символов, а передан параметр, состоящий из большего или меньшего количества символов то, ошибка: "Неверный формат поля &lt;Наименования полей через запятую&gt;", запись не производится.</li> </ul> </li> <li>– если запись на переданную в методе бирку для данного пациента найдена в БД, то ошибка "Человек уже записан на данную бирку", запись не производится.</li> </ul> <p>Выходной параметр для случаев, когда бирка, на которую пытаются записаться, заблокирована:</p> <ul style="list-style-type: none"> <li>– TimeTableStatusLockComment (Т, Н) – сообщение о блокировке бирки. Сообщение содержит в себе возможное время разблокировки бирки.</li> </ul> <p>Проверка на наличие переданной во входных параметрах бирки в таблице БД по TimeTableMedService_id:</p> <ul style="list-style-type: none"> <li>– 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;</li> </ul>

	<ul style="list-style-type: none"> <li>– в связи с данной проверкой для бирок, которые нашлись в таблице БД выводится в ответе параметр: <ul style="list-style-type: none"> <li>– 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).</li> </ul> </li> </ul> <p>Успешный ответ – составной идентификатор записи на прием:</p> <ul style="list-style-type: none"> <li>– Person_id (N, O), TimeTableGraf_id (N, O) – составной идентификатор записи на прием.</li> </ul> <p>Если во входящих параметрах указан идентификатор постановки в очередь, то в направление передается TimeTableGraf_id, а Evn_Queue_id удаляется.</p>
--	--

## **5.8 Методы сервиса "Подтверждение записи на прием к врачу с возможностью отмены и переноса (исходящий обзвон)"**

### **5.8.1 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.

Таблица 28 – Параметры метода **GET api/TimeTableList**

<b>Входящие параметры</b>	<ul style="list-style-type: none"> <li>– Lpu_id (N, H) – идентификатор медицинской организации;</li> <li>– TimeTable_beg (DT, H) – дата и время начала диапазона поиска (если не указано, то считать текущие время и дату за дату начала диапазона);</li> <li>– MedStaffFact_id (N, H) – идентификатор места работы. Правила передачи: <ul style="list-style-type: none"> <li>– 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;</li> <li>– 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;</li> </ul> </li> <li>– Post_id (N, H) – идентификатор должности. Правила передачи: <ul style="list-style-type: none"> <li>– если передано значение и в параметре TimeTableSource</li> </ul> </li> </ul>
---------------------------	---

	<p>передано значение "MedService", то метод вернёт бирки в соответствии с указанным типом (на службу), переданные значения параметра Post_id будут проигнорированы;</p> <ul style="list-style-type: none"> <li>– 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.</li> <li>– TimeTable_end (DT,H) – дата и время окончания диапазона;</li> <li>– TimeTableSource (T, H) – тип бирки. Возможные значения: Graf/MedService для получения бирок на прием к врачу/на услугу службы соответственно;</li> <li>– offset (N,H,0) – номер начальной возвращаемой записи;</li> <li>– limit (N,H,100) – максимальное число возвращаемых записей;</li> <li>– Page (N,H,1) – номер страницы.</li> </ul>
<p><b>Ответ</b></p>	<p>Успешный ответ – список бирок, на которые записаны пациенты, в заданной МО за заданный диапазон дат.</p> <p>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.</p> <p>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.</p> <p>Если у УЗ пользователя нет группы "Вендор API", то в ответе будут переданы записи только по МО пользователя.</p> <p>Бирки в ответе выводить по степени близости приема к TimeTable_beg.</p> <p>Возвращаемые параметры:</p> <ul style="list-style-type: none"> <li>– Lpu_id (N, O) – идентификатор МО;</li> <li>– Org_Name (T, O) – Наименование МО;</li> <li>– TimeTables(S, O) – объект:</li> <li>– Person_id (N, O) – идентификатор пациента;</li> <li>– Person_SurName (T, O) – фамилия пациента;</li> <li>– Person_FirName (T, O) – имя пациента;</li> <li>– Person_SecName (T, O) – отчество пациента;</li> <li>– Person_Phone (T, O) – номер телефона пациента;</li> <li>– TimeTableSource (T, O) – тип бирки:Graf или MedService;</li> <li>– TimeTable_id (N, O) – идентификатор бирки;</li> <li>– TimeTableType_id (N, O) – тип бирки;</li> <li>– TimeTable_factTime (DT, H) – фактические дата и время приема для обслуженных бирок (если у бирки есть Фактическое время</li> </ul>

	<p>приема TimeTableGraf_factTime/TimeTableMedService_factTime, то пациент приходил);</p> <ul style="list-style-type: none"> <li>– TimeTable_begTime (DT, O) – дата и время начала приёма;</li> <li>– MedPersonal_id (N) – идентификатор сотрудника;</li> <li>– MedPersonal_SurName (T) – фамилия сотрудника;</li> <li>– MedPersonal_FirName (T) – имя сотрудника;</li> <li>– MedPersonal_SecName (T) – отчество сотрудника;</li> <li>– MedPersonal_FIO (T) – Ф. И. О. сотрудника в нормализованном виде.</li> <li>– MedService_id (N, H) – идентификатор службы (выводить только если тип бирки MedService);</li> <li>– MedService_Name (T, H) – наименование службы (выводить только если тип бирки MedService);</li> <li>– MedStaffFact_id (N, H) – идентификатор врача (выводить только если бирка поликлиническая Graf);</li> <li>– Post_id (N, H) – идентификатор должности врача (выводить только если бирка поликлиническая Graf);</li> <li>– Post_name (T, H) – наименование должности врача (выводить только если бирка поликлиническая Graf);</li> <li>– LpuSectionProfile_id (N, O) – идентификатор профиля отделения;</li> <li>– Usluga_Code (T, H) – код услуги;</li> <li>– Usluga_Name (T, H) – наименование услуги.</li> </ul>
<b>Возможные ошибки</b>	<ul style="list-style-type: none"> <li>– код "1", прикладной код "6", текст ошибки: "Неверное значение TimeTableSource. Должно быть Graf или MedService ";</li> <li>– код "2", прикладной код "6", текст ошибки: "Значение MedStaffFact_id или Post_id не найдено в системе"</li> </ul>

### 5.8.2 Получение списка предстоящих приемов пациента в МО GET api/TimeTableListbyPatient

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

Таблица 29 – Параметры метода **GET api/TimeTableListbyPatient**

<b>Входящие параметры</b>	<ul style="list-style-type: none"> <li>– Person_id (N,O) – идентификатор пациента;</li> <li>– TimeTable_beg (DT,H) – дата и время начала диапазона поиска</li> </ul>
---------------------------	--

	<p>(если не указано, то считать текущие время и дату за дату начала диапазона);</p> <ul style="list-style-type: none"> <li>– TimeTable_end (DT,H) – дата и время окончания диапазона;</li> <li>– Lpu_id (N,H) – идентификатор медицинской организации (доступно только для вендора)</li> </ul>
<b>Ответ</b>	<p>Правила:</p> <ul style="list-style-type: none"> <li>– значение параметра TimeTable_beg не может быть ранее текущих даты и времени. Иначе метод вернёт ошибку;</li> <li>– если параметр TimeTable_beg не задан, то устанавливается равным текущим дате и времени;</li> <li>– значение параметра TimeTable_end не может быть меньше параметра TimeTable_beg. Иначе метод вернёт ошибку;</li> <li>– если идентификатор человека не найден в Системе, то метод вернёт ошибку.</li> </ul> <p>Возвращаемые параметры:</p> <ul style="list-style-type: none"> <li>– Person_id (N, O) – идентификатор пациента;</li> <li>– Person_SurName (T, O) – фамилия пациента;</li> <li>– Person_FirName (T, O) – имя пациента;</li> <li>– Person_SecName (T, O) – отчество пациента;</li> <li>– Person_Phone (T, O) – номер телефона пациента;</li> <li>– TimeTable (S, O) – список ячеек расписания, на которые записан человек: <ul style="list-style-type: none"> <li>– Lpu_id (N, O) – идентификатор медицинской организации (доступно только для вендора);</li> <li>– TimeTableSource (T, O) – метка источника данных;</li> <li>– TimeTable_id (N, O) – идентификатор ячейки расписания;</li> <li>– 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);</li> <li>– TimeTable_begTime (T, O) – дата и время начала приёма;</li> <li>– MedService_id (N, H) – идентификатор службы;</li> <li>– MedService_Name (T, H) – наименование службы;</li> <li>– MedStaffFact_id (N, H) – идентификатор места работы врача;</li> <li>– Post_id (N, O) – идентификатор должности врача;</li> <li>– Post_name (T, O) – наименование должности врача;</li> <li>– LpuSectionProfile_id (N, O) – идентификатор профиля отделения;</li> <li>– Usluga_Code (T, H) – код услуги;</li> <li>– Usluga_Name (T, H) – наименование услуги.</li> </ul> </li> </ul>

### 5.8.3 Изменение статуса записи на прием PUT api/TimeTableGraf/TimeTableGrafStatus

Таблица 30 – Параметры метода PUT api/TimeTableGraf/TimeTableGrafStatus

<b>Входящие параметры</b>	<ul style="list-style-type: none"> <li>– Person_id (N, O), TimeTableGraf_id (N, O) – составной идентификатор записи на прием.</li> <li>– EvnStatus_id (N,O) – идентификатор статуса направления</li> <li>– TimeTableGraf_id (N, O) – ссылка на бирку поликлиники</li> </ul>
<b>Ответ</b>	<p>Успешный ответ – пустой ответ с кодом ошибки "0"</p> <p>Если EvnStatus_id меняется на 12 или 13, то в таблице dbo.TimeTableGraf значения полей RecClass_id, Person_id, EvnDirection_id меняется на NULL.</p>

## 5.9 Методы сервиса "Вызов врача на дом"

### 5.9.1 Добавление вызова врача на дом POST api/HomeVisit/HomeVisit

Таблица 31 – Параметры метода POST api/HomeVisit/HomeVisit

<b>Входящие параметры</b>	<ul style="list-style-type: none"> <li>– Person_id (N, O) – идентификатор пациента; если пациент не найден, тогда метод вернет ошибку "Пациент с ИД [%s] не найден".</li> <li>– CallProfType_id (N, O) – идентификатор профиля вызова;</li> <li>– Address_Address (T, O) – адрес вызова;</li> <li>– HomeVisitCallType_id (N, O) – идентификатор типа вызова;</li> <li>– HomeVisit_setDT (DT, O) – дата и время вызова;</li> <li>– 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;</li> <li>– MedStaffFact_id (N, H) – место работы врача;</li> <li>– HomeVisit_Phone (T, H) – телефон обратной связи;</li> <li>– HomeVisitWhoCall_id (N, O) – значение из справочника "Кто вызвал" <a href="http://fixuploader.swn.local/dbspr/index/109620575">http://fixuploader.swn.local/dbspr/index/109620575</a>;</li> <li>– HomeVisit_Symptoms (T, O) – симптомы;</li> <li>– HomeVisit_Comment (T, H) – дополнительная информация;</li> <li>– HomeVisitStatus_id (N, O) – идентификатор статуса вызова;</li> <li>– 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</li> </ul>
---------------------------	--

	<p>ошибку; если у улицы с таким идентификатором нет GUID (т.е. кода нет в БД, не привязан), то метод вернёт ошибку;</p> <ul style="list-style-type: none"> <li>– HomeVisit_StreetGUID (Т, О) – 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";</li> <li>– KLAdr_Actual = 0 (N, H) – фильтр актуальности данных;</li> <li>– HomeVisit_House (Т, О) – номер дома;</li> <li>– HomeVisit_LpuComment (Т, Н) – причина отказа. Обязательное, если HomeVisitStatus = "Отказ";</li> <li>– Slot_Id (N, H) – идентификатор "Виртуальной бирки" вызова на дом;</li> <li>– LpuRegion_cid (N, H) – участок вызова;</li> <li>– HomeVisit_IsQuarantine (N, H) – признак "Карантин". Возможные значения: 1 и 2, где 1 - нет, 2 – да;</li> <li>– Report (N, H) – код отчёта</li> <li>– 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)</li> </ul>
<p><b>Ответ</b></p>	<p>Успешный ответ</p> <p>Код ошибки – 0</p> <ul style="list-style-type: none"> <li>– HomeVisit_id (N) – идентификатор вызова на дом;</li> <li>– HomeVisitStatus_id (N, H) – идентификатор статуса вызова. Значение из справочника dbo.HomeVisitStatus;</li> <li>– HomeVisitStatus_name (Т, Н) – наименование статуса вызова. Значение из справочника dbo.HomeVisitStatus;</li> <li>– Slot_Id (N, H) – идентификатор "виртуальной бирки" вызова на дом.</li> </ul>

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:
- Если указан то переход к п. 15;
  - 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:
- Если оно найдено, тогда найденный врач устанавливается, как врач на вызов и переход к п. 16.
  - Иначе ошибка "Указанное место работы MedstaffFact\_id на участке не найдено"
- 16) Проверяется наличие профиля отделения участка:
- Если профиль отделения отсутствует, тогда ошибка – "Не удалось найти МО, обслуживающую указанный адрес. Профиль отделения не найден. Ошибка БД."
  - Иначе переход к п.17.
- 17) Проверяется статус профиля отделения – "Работает по участковому принципу":
- 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")
  - Иначе переход к п.18.
- 18) Проверяется группа отделений к которой относится найденное отделение:
- Если группа отделений не найдена, тогда ошибка – "Не удалось найти МО, обслуживающую указанный адрес. Группа отделений не найдена. Ошибка БД.";
  - Иначе переход к п.19.
- 19) Проверяется тип группы отделений к которому относится найденное отделение:
- Если тип группа отделений НЕ "Поликлиника", тогда ошибка – "Не удалось найти МО, обслуживающую указанный адрес. Тип группы отделений должен быть [Поликлиника]."
  - 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:
- Если вызов не найден, тогда переход к п. 22.
  - Иначе ошибка – "Для данного пациента уже имеется необслуженный вызов врача на дом".
- 21) Генерируется номер вызова автоматически – плюс 1 к последнему номера вызова:
- 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": "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;

	<ul style="list-style-type: none"> <li>– По указанной улице должна быть обслуживающая МО;</li> <li>– По указанной улице и дому должен быть обслуживающий участок;</li> <li>– Тип участка должен соответствовать возрасту пациента;</li> <li>– 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;</li> <li>– Тип группы отделений, к которому относится участок должен быть "Поликлиника";</li> <li>– Если указывается LpuRegion_cid, тогда он должен присутствовать в МО, обслуживающей указанный адрес;</li> <li>– Если указывается LpuRegion_cid, тогда тип участка должен соответствовать возрасту пациента;</li> <li>– Если указывается MedStaffFact_id, тогда он должен присутствовать на участке, обслуживающей указанный адрес или на участке LpuRegion_cid.</li> <li>– Если требуется автоматическое назначение врача на вызов, тогда требуется указание входного параметра HomeVisitStatus_id=6 (Назначен врач).</li> </ul> <p><b>Успешный ответ:</b></p> <p>Код ошибки – 0</p> <ul style="list-style-type: none"> <li>– HomeVisit_id (N) – Идентификатор вызова на дом</li> <li>– HomeVisitStatus_id (N) – Идентификатор статуса вызова. Значение из справочника dbo.HomeVisitStatus</li> <li>– HomeVisitStatus_name (T) – Наименование статуса вызова. Значение из справочника dbo.HomeVisitStatus</li> <li>– Slot_Id (N) – Идентификатор "виртуальной бирки" вызова на дом. При создании вызова на дом значение Slot_Id сохраняется в dbo.HomeVisit.HomeVisit_GUID</li> <li>– HomeVisit_setDT (DT) – Дата и время вызова</li> <li>– Duration (N) – Продолжительность приема. По умолчанию передается значение 60 (в минутах)</li> </ul>
<b>Ошибки</b>	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

	<p>пациента возрастом меньше 18 лет.</p> <ul style="list-style-type: none"> <li>– Вызов врача на дом может быть оформлен через портал в рабочие дни поликлиники %s.</li> <li>– Невозможно сгенерировать нумератор.</li> <li>– Невозможно сгенерировать номер вызова.</li> <li>– Невозможно создать вызов на дом на умершего человека.</li> <li>– Ошибка запроса к БД.</li> <li>– Указанный тип участка LpuRegion_cid не соответствует возрасту пациента.</li> <li>– Тип участка указанного LpuRegion_cid некорректный.</li> <li>– Указанное место работы MedstaffFact_id на участке не найдено.</li> </ul>
<p><b>Пример</b></p>	<p>Запрос:</p> <p>POST</p> <p>http://41484.prm.bot8.promedweb.ru/api/HomeVisit/HomeVisit?CallProfType_id=1&amp;Address_Address=РОССИЯ, ПЕРМСКИЙ КРАЙ, ПЕРМЬ Г, УРАЛЬСКАЯ УЛ, д. 2, кв. 64&amp;HomeVisitCallType_id=1&amp;HomeVisit_setDT=30.03.2021 09:58&amp;HomeVisitWhoCall_id=1&amp;Person_id=100293729&amp;HomeVisit_StreetGUID=68b ff86d-e5cb-46e6-bc78-4dee516fbc88&amp;HomeVisit_Symptoms=Б/Л: да&amp;HomeVisitStatus_id=1&amp;HomeVisit_House=2</p> <p>Ответ:</p> <p>Ошибка</p> <pre>{   "error_code": 0,   "data": [     {       "HomeVisit_id": "590101000000223"     }   ] }</pre>

	] }
--	--------

## 5.9.2 Получение адреса GET api/Address

Таблица 32 – Параметры метода GET api/Address

<b>Входящие параметры</b>	<ul style="list-style-type: none"> <li>– Person_id (N, H) – идентификатор человека;</li> <li>– Address_id (N, H) – идентификатор адреса;</li> <li>– AddressType_id (N, H) – тип адреса (1 – Адрес регистрации, 2 – Адрес проживания, 3 – Адрес рождения);</li> <li>– Area_AOGUID (T, H) – GUID территории (города, населенного пункта и т.п.) по ФИАС;</li> <li>– Street_AOGUID (T, H) – GUID улицы по ФИАС</li> </ul>
<b>Ответ</b>	<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"> <li>– Person_id (N, H) – идентификатор человека;</li> <li>– Address_id (N, O) – идентификатор адреса;</li> <li>– AddressType_id (N, H) – тип адреса (1 – Адрес регистрации, 2 – Адрес проживания, 3 – Адрес рождения);</li> <li>– Address_Zip (S[10], H) – индекс;</li> <li>– KLCountry_id (N, O) – страна;</li> <li>– KLRgn_id (N, Y) – регион;</li> <li>– KLSubRgn_id (N, Y) – район;</li> <li>– KLCity_id (N, Y) – город;</li> <li>– KLTown_id (N, Y) – нас. пункт;</li> </ul>

	<ul style="list-style-type: none"> <li>– KLStreet_id (N, O) – улица;</li> <li>– Address_House (N, O) – дом;</li> <li>– Address_Corpus (N, H) – корпус;</li> <li>– Address_Flat (N, H) – квартира;</li> <li>– Address_Address (T, H) – полный адрес в текстовом формате;</li> <li>– Address_insDT (D, O) – дата изменения записи адреса в формате ГГГГ–ММ–ДД;</li> <li>– AoidArea (T[36],Y) – идентификатор адресного объекта (населенного пункта, города) по ФИАС – значение справочника KLArea, поле KLArea_AOID;</li> <li>– AoidStreet (T[36],Y) – идентификатор адресного объекта (улицы) по ФИАС – значение справочника KLStreet, поле KLStreet_AOID;</li> <li>– KLAreaType_id (N, H) – идентификатор справочника тип территории;</li> <li>– Address_insDatetime (DT, H) – дата и время записи об адресе;</li> <li>– KLCountry_name (T, O) – наименование страны;</li> <li>– KLRgn_name (T, Y) – наименование региона;</li> <li>– KLSubRgn_name (T, Y) – наименование района;</li> <li>– KLCity_name (T, Y) – наименование города;</li> <li>– KLTown_name (T, Y) – наименование населенного пункта;</li> <li>– KLStreet_name (T, Y) – наименование улицы.</li> </ul> <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"> <li>– код состояния: 200</li> <li>– значение параметров: <ul style="list-style-type: none"> <li>– error_code: 23001</li> <li>– error_msg: Отсутствует обязательный параметр (наименование параметра)</li> </ul> </li> </ul>
<b>Пример</b>	<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>
--	---

### 5.9.3 Получение списка параметров адреса GET api/AddressSearch

Метод возвращает список параметров адреса.

Таблица 33 – Параметры метода GET api/AddressSearch

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

	<p>области/края/республики и т.д.</p> <ul style="list-style-type: none"> <li>– Area_AOGUID (T) – идентификатор адресного объекта по ФИАС (области/края/республики и т.д.).</li> <li>– SubRegion_id (N) – идентификатор района.</li> <li>– SubRegion_Name (T) – Наименование района.</li> <li>– SubRegion_Type (T) – Тип района.</li> <li>– SubRegion_KLADR_Code (T) : Код КЛАДР района.</li> <li>– SubRegion_Postal_Index (T) – Почтовый индекс района.</li> <li>– SubRegion_AOGUID (T) – идентификатор адресного объекта по ФИАС (района).</li> <li>– City_id (N) – идентификатор города.</li> <li>– City_Name (T) – Наименование населённого пункта (города, деревни и т.п.)</li> <li>– City_Type (T) – Тип населённого пункта.</li> <li>– City_KLADR_Code (T) – Код КЛАДР населённого пункта.</li> <li>– City_Postal_Index (T) – Почтовый индекс населённого пункта.</li> <li>– City_AOGUID (T) – идентификатор адресного объекта по ФИАС (населённого пункта).</li> <li>– Street_id (N) – идентификатор улицы.</li> <li>– Street_Name (T) – Наименование улицы (переулка, бульвара и т.п.).</li> <li>– Street_Type (T) – Тип улицы.</li> <li>– Street_KLADR_Code (T) – Код КЛАДР улицы.</li> <li>– Street_Postal_Index (T) – Почтовый индекс улицы.</li> <li>– Street_AOGUID (T) – идентификатор адресного объекта по ФИАС (улицы).</li> <li>– House (T) – Номер дома (строения, здания и т.п.).</li> <li>– House_Corpus (T) – Номер корпуса.</li> <li>– House_Type (T) – Тип дома.</li> <li>– House_Postal_Index (T) – Почтовый индекс дома.</li> <li>– Flat (T) – Номер квартиры.</li> </ul>
--	--

	<p>Правила:</p> <ul style="list-style-type: none"> <li>– Если найден один адрес, то метод вернёт параметры адреса.</li> <li>– Если не найдено адреса, то метод вернёт пустое значение.</li> <li>– Если найдено более одного адреса, то метод вернёт ошибку.</li> <li>– Если строка адреса не соответствует допустимым форматам, то метод вернёт ошибку.</li> <li>– В качестве сокращений части адреса допустимы любые из справочника dbo.KLSocг с учётом уровня/номера части адреса.</li> <li>– Сокращения могут быть в начале части адреса и в конце.</li> <li>– Сокращения могут отсутствовать.</li> <li>– Корпус дома берётся из строки адреса.</li> <li>– Номер квартиры берётся из строки адреса.</li> </ul>
--	---

#### 5.9.4 Метод добавления талона вызова POST /emergency/EmergencyCallCard

Метод предназначен для добавления талона вызова

Таблица 34 – Параметры метода **POST /emergency/EmergencyCallCard**

<b>Входящие параметры</b>	<ul style="list-style-type: none"> <li>– CmpCallCard – талон вызова: <ul style="list-style-type: none"> <li>– Guid (T);</li> <li>– primaryCallCard.id (long) – ссылка на первичный вызов;</li> <li>– number (long, Y) – номер вызова за день;</li> <li>– numberYear (long, Y) – номер вызова за год;</li> <li>– priority (N) – приоритет;</li> <li>– region.id (long) – регион адреса вызова (KLArea);</li> <li>– subRegion.id (long) – район адреса вызова (KLArea);</li> <li>– city.id (long) – город адреса вызова (KLArea);</li> <li>– town.id (long) – населенный пункт адреса вызова (KLArea);</li> <li>– street.id (long) – улица адреса вызова (KLStreet);</li> <li>– streetSecond.id (long) – 2-я улица перекрестка (KLStreet);</li> </ul> </li> </ul>
---------------------------	---

	<ul style="list-style-type: none"> <li>– house (Т (30)) – номер дома адреса;</li> <li>– корпус (Т (5)) – номер корпуса адреса;</li> <li>– flat (Т (5)) – номер квартиры;</li> <li>– room (Т (5)) – номер комнаты;</li> <li>– entrance (Т (200)) – номер подъезда. Размерность параметра 200 символов;</li> <li>– floor (Т (200)) – номер этажа. Размерность параметра 200 символов;</li> <li>– entranceCode (Т (200)) – код домофона. Размерность параметра 200 символов;</li> <li>– phone (Т (20)) – номер телефона;</li> <li>– comment (Т (4000)) – комментарий;</li> <li>– 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;</li> <li>– 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</li> <li>– person.id (long, O) – идентификатор человека. Person_id (для неидентифицированных пациентов создается неизвестный человек);</li> <li>– surname (Т (30)) – фамилия пациента;</li> <li>– firname (Т (30)) – имя;</li> <li>– secname (Т (30)) – отчество;</li> <li>– age (N) – возраст;</li> <li>– ageDays (N) – возраст (дней);</li> <li>– ageMonths (N) – возраст (полных месяцев);</li> <li>– birthday (S (DT) (23)) – дата рождения;</li> <li>– sex.id (N) – идентификатор половой принадлежности (Sex);</li> <li>– polis.series (Т (10)) – серия полиса ОМС;</li> <li>– polis.number (Т (30)) – номер полиса ОМС;</li> <li>– polis.federalNumber (Т (16)) – единый номер полиса ОМС;</li> <li>– whoCallType.id (long) – вид вызывающего помощь;</li> <li>– whoCall (Т (30)) – кто вызывает помощь;</li> <li>– callType.id (long) – тип вызова (CmpCallType);</li> <li>– callProfile.id (long) – профиль вызова (CmpProfile);</li> <li>– urgency (N) – срочность вызова;</li> <li>– acceptTime (Т (DT) (23), Y) – дата и время получения вызова;</li> <li>– endingTime (Т (DT) (23), Y) – дата и время окончания вызова;</li> <li>– arrivalPlanTime (Т (DT) (23)) – плановое время прибытия на место вызова;</li> </ul>
--	---

	<ul style="list-style-type: none"> <li>- arrivalTime (T (DT) (23)) – время прибытия на место вызова;</li> <li>- hospitalArrivalTime (T (DT) (23)) – время прибытия в стационар;</li> <li>- transportTime (string (datetime) (23)) – время начала транспортировки;</li> <li>- departureTime (string (datetime) (23)) – время выезда на вызов;</li> <li>- acceptLine (integer) – пульт приема;</li> <li>- callDuration (integer) – длительность приема вызова (сек);</li> <li>- emergencyTeam.id (long, Y) – идентификатор бригады вызова (EmergencyTeam);</li> <li>- emergencyTeamSpec.id (long) – идентификатор профиля бригады СМП;</li> <li>- emergencyTeamNum (string (32)) – номер бригады;</li> <li>- medstafTeam.id (long, Y) – врач бригады СМП;</li>   <li>- transferTime (string (datetime) (23)) – время передачи вызова;</li> <li>- lpu.id (long, Y) – МО обслуживания вызова (Lpu);</li> <li>- lpuAccept.id (long) – МО принятия вызова (Lpu);</li> <li>- lpuTransferUrgent.id (long) – МО передачи вызова (НМП);</li> <li>- lpuTransferEmergency.id (long) – МО передачи вызова (СМП);</li> <li>- lpuBuilding.id (long) – подразделение СМП (LpuBuilding);</li> <li>- lat (number) – широта адреса вызова;</li> <li>- lon (number) – долгота адреса вызова;</li> <li>- diag.id (long, Y) – основной диагноз;</li> <li>- diag.code (string (10),Y) – код диагноза по МКБ;</li>   <li>- isUrgent (boolean) – признак неотложной помощи;</li> <li>- isOpen (Boolean) – признак открытой карты;</li> <li>- isActiveToPolyclinic (Boolean) – признак актива в поликлинику;</li> <li>- callPlaceType.id (long) – тип места вызова (CmpCallPlaceType);</li> <li>- nsiCmpCallPlaceType.id (long) – место оказания СМП;</li> <li>- lpuHospitalization.id (long, Y) – МО госпитализации (Lpu);</li> <li>- address.fiasguid (string (uniqueidentifier)) – ГУИД кода ФИАС адреса места вызова</li> <li>- address.fiascode (string (33),Y) – код ФИАС адреса места вызова;</li> <li>- addressTxt (string(1000), N) – текстовое описание адреса вызова;</li> <li>- commonState.id (long) – идентификатор степени тяжести состояния;</li> </ul>
--	---

	<ul style="list-style-type: none"> <li>– callSignType.code (integer) – тип вызова СМП;</li> <li>– stationNum (integer) – подстанция СМП (номер);</li> <li>– transportation.features (String) – особенности, связанные с транспортировкой и оказанием скорой медицинской помощи больному;</li> <li>– numberCS (string (255)) – номер сопроводительного листа;</li> <li>– document.number (string(30), Н) – серия и номер документа удостоверяющего личность;</li> <li>– document.orgDepText (string(1000), Н) – организация, выдавшая документ удостоверяющий личность;</li> <li>– document.begDate (datetime, Н) – дата выдачи документа удостоверяющего личность;</li> <li>– uaddress (string (1000), Н) – адрес места жительства пациента;</li> <li>– accidentCircumstance (string(512), Н) – обстоятельства несчастного случая;</li> <li>– SmpEmTeamDuty: <ul style="list-style-type: none"> <li>– emergencyTeamWatch.id (long) – дежурство бригады СМП;</li> </ul> </li> <li>– 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)</li> </ul>
<p><b>Ответ</b></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  
}
```

## 5.10 Методы сервиса "Запись на вакцинацию"

### 5.10.1 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.

Таблица 35 – Параметры метода GET api/VaccinationTimeTableMedService

<b>Запрос</b>	<ul style="list-style-type: none"><li>– Person_id (N,H) – идентификатор человека. Параметр обязателен если не указаны параметры KLStreet_id и Address_House, или не указан параметр Address_Address;</li><li>– VaccinationType_Id (N,O) – тип прививки. Запрашивается значение справочника "Типы прививок ФЭР";</li><li>– KLStreet_id (N, H) – идентификатор улицы проживания. Параметр обязателен если не указан параметр Person_id или Address_Address;</li><li>– Address_House (T, H) – номер дома;</li><li>– TimeTable_begDT (D, H) – начало диапазона поиска свободных ячеек расписания;</li><li>– TimeTable_endDT (D, H) – конец диапазона поиска свободных ячеек расписания;</li><li>– Address_Address (T, H) – строка адреса в формализованном виде:<ul style="list-style-type: none"><li>– Формат строки: индекс, страна, область, район, город/населённый пункт, улица, номер дома, номер корпуса, номер квартиры;</li><li>– допустимы форматы:,,,город,улица,номер дома/здания,,;</li></ul>Указанные параметры обязательны. Ведущие/начальные запятые обязательны</li><li>– LpuSectionAge_Code (N, H) – код типа отделения по возрасту (предназначен для определения возрастной группы прививаемого в случае поиска бирки только по параметру Address_Address)</li></ul>
---------------	---

<b>Ответ</b>	<p>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).</p> <p>Если не задан параметр VaccinationType_id – ошибка: не указан тип вакцинации.</p> <ul style="list-style-type: none"> <li>– VaccinationType_Id (N, H) – тип прививки. Значение справочника "Типы прививок ФЭР";</li> <li>– VaccinationType_Name (T, H) – наименование прививки;</li> <li>– Lpu_id (N, O) – идентификатор МО удовлетворяющего условиям поиска;</li> <li>– Services(S, O): <ul style="list-style-type: none"> <li>– MedService_id (N, O) – идентификатор службы;</li> <li>– MedService_Name (T, O) – имя службы;</li> <li>– 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.</li> </ul> </li> <li>– 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`";</li> <li>– LpuBuildingOffice_Number (T, O) – номер кабинета;</li> <li>– TimeTable(S, O): <ul style="list-style-type: none"> <li>– TimeTable_id (N, O) – идентификатор бирки;</li> <li>– TimeTable_begTime (DT, O) – дата и время начала приёма в формате ГГГГ-ММ-ДД ЧЧ:ММ:СС</li> </ul> </li> </ul>
--------------	---

### 5.10.2 Запись пациента на приём службы POST api/TimeTableMedService/TimeTableMedServiceWrite

Таблица 36 – Параметры метода POST api/TimeTableMedService/TimeTableMedServiceWrite

<b>Входящие параметры</b>	<ul style="list-style-type: none"> <li>– Person_id (N, O) – идентификатор пациента;</li> <li>– MedService_id (N, Y) – идентификатор службы. Является обязательным, если параметр UslugaComplexMedService_id пустой;</li> <li>– UslugaComplexMedService_id (N, Y) – идентификатор связи комплексной услуги и службы. Является обязательным, если параметр MedService_id пустой;</li> <li>– TimeTableMedService_id (N, O) – идентификатор свободной бирки в расписании службы;</li> <li>– EvnQueue_id (N, H) – идентификатор постановки в очередь. Выходные параметры: <ul style="list-style-type: none"> <li>– Person_id (N, O) – идентификатор пациента;</li> <li>– TimeTableMedService_id (N, O) – идентификатор записи на службу/услугу службы;</li> </ul> </li> <li>– EvnQueue_id (N, H) – идентификатор постановки в очередь</li> </ul>
---------------------------	---

<p><b>Ответ</b></p>	<ul style="list-style-type: none"> <li>- Person_id (N, O) – идентификатор пациента;</li> <li>- TimeTableMedService_id (N, O) – идентификатор записи на службу/услугу службы;</li> <li>- EvnQueue_id (N, H) – идентификатор постановки в очередь;</li> <li>- TimeTableStatusLockComment (Т,Н) – Сообщение о блокировке бирки, отображается в ответе если бирка присутствует в БД. Сообщение содержит в себе возможное время разблокировки бирки.</li> </ul> <p>Правила:</p> <ul style="list-style-type: none"> <li>- если во входящих параметрах указан идентификатор постановки в очередь: <ul style="list-style-type: none"> <li>- в направлении передается TimeTableMedService_id;</li> <li>- Evn_Queue_id удаляется.</li> </ul> </li> <li>- 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";</li> <li>- 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 &lt;указывается MedService_id или UslugaComplexMedService_id &gt; не верно указан идентификатор бирки ";</li> <li>- если запрос обработан успешно, то в ответе должно быть отправлено: <ul style="list-style-type: none"> <li>- код состояния: 200;</li> <li>- параметры: <ul style="list-style-type: none"> <li>- error_code: 0;</li> <li>- data: <ul style="list-style-type: none"> <li>- Person_id;</li> <li>- TimeTableMedService_id;</li> <li>- EvnQueue_id.</li> </ul> </li> </ul> </li> </ul> </li> </ul> <p>Проверка на наличие переданной во входных параметрах бирки в таблице БД по TimeTableMedService_id:</p> <ul style="list-style-type: none"> <li>- 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;</li> <li>- в связи с данной проверкой для бирок, которые нашлись в таблице БД выводится в ответе параметр:</li> <li>- 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)</li> </ul>
---------------------	--

## **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.

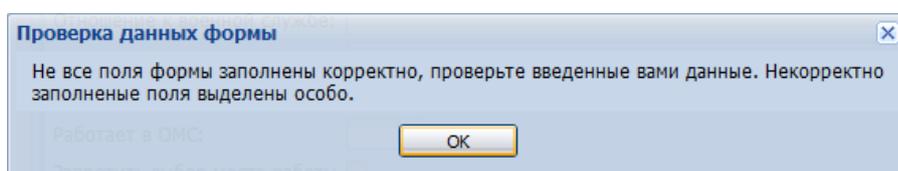
## **6.2 Действия в случае несоблюдения условий выполнения технологического процесса**

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.