

ООО "РТ МИС"

ЕДИНАЯ ЦИФРОВАЯ ПЛАТФОРМА.МИС 3.0

(ЕЦП.МИС 3.0)

Руководство администратора

Подсистема "Взаимодействие с внешними голосовыми ботами (в т.ч. технологии ИИ) 122" 3.0.3_1

Сервис "Запись к врачу" 3.0.3_1

Содержание

1	Введение	4
1.1	Область применения	4
1.2	Краткое описание возможностей	4
1.3	Уровень подготовки администратора.....	4
1.4	Перечень эксплуатационной документации, с которым необходимо ознакомиться администратору	4
2	Общие сведения	5
2.1	Описание информационного взаимодействия.....	5
2.2	Схема взаимодействия	5
2.3	Кэширование запросов.....	5
2.4	HTTP ошибки.....	5
2.4.1	Ошибки на стороне клиента.....	5
2.4.2	Ошибки на стороне сервера	6
2.4.3	Детализация ошибок	6
3	Авторизация	7
3.1	Авторизация в PHP API. Авторизация в системе.....	7
3.2	Авторизация в SWAN API. Метод авторизации пользователя СМП.....	8
4	Описание сценария сервиса "Запись к врачу".....	9
5	Описание методов.....	12
5.1	Общие параметры для всех методов	12
5.2	Описание параметров метода.....	12
5.3	Описание общих кодов ошибок	13
5.4	Описание методов работы с человеком	13
5.4.1	Получение списка физических лиц, по ключевым параметрам GET api/PersonList.....	13
5.4.2	Получение информации по человеку GET api/Person	14
5.4.3	Создание человека POST api/Person	16
5.4.4	Редактирование данных человека PUT api/Person	21
5.4.5	Получение записей на прием к врачу GET api/TimeTableGraf/TimeTableGrafByMedStaffFact	25
5.4.6	Получение прикреплений пациента GET api/PersonAttach.....	25
5.4.7	Получение основного прикрепления пациента GET api/PersonMainAttach	26
5.5	Общие методы.....	29
5.5.1	Получение элементов справочника GET api/Refbook	29

5.5.2	Поиск человека по атрибутам GET /api/Person/mgetPersonSearch	31
5.6	Методы работы с листами ожидания.....	36
5.6.1	Добавление записи в лист ожидания POST api/EvnQueue/EvnQueue	36
5.7	Методы сервиса "Запись к врачу"	37
5.7.1	Получение должности по наименованию GET api/PostByName	37
5.7.2	Получение МО по идентификатору GET api/MOById	37
5.7.3	Получение информации о докторе GET /api/doctors/MedStaffFact_id	37
5.7.4	Получение списка специальностей в МО GET api/MedSpecOms/MedSpecOmsByMO	41
5.7.5	Получение списка мест работы по специальности и МО, или по должности и МО GET api/MedStaffFact/MedStaffFactByMO	41
5.7.6	Получение свободных дат приема GET api/TimeTableGraf/TimeTableGrafFreeDate.....	44
5.7.7	Получение свободного времени приема GET api/TimeTableGraf/TimeTableGrafFreeTime.....	44
5.7.8	Запись пациента на прием POST api/TimeTableGraf/TimeTableGrafWrite.....	46
5.7.9	Получение списка прошедших и предстоящих занятых бирок в МО за заданный временной промежуток GET api/TimeTableList	47
5.7.10	Получение списка предстоящих приемов пациента в МО GET api/TimeTableListbyPatient	50
5.7.11	Изменение статуса записи на прием PUT api/TimeTableGraf/TimeTableGrafStatus	51
6	Аварийные ситуации	52
6.1	Описание аварийных ситуаций.....	52
6.2	Действия в случае несоблюдения условий выполнения технологического процесса ..	53

1 Введение

1.1 Область применения

Настоящий документ описывает порядок работы с сервисом "Запись к врачу" 3.0.3_1 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.

Сервис "Запись к врачу" обеспечивает запись на приём к врачу пациента без направления с помощью голосового помощника с возможностью записи в лист ожидания

1.3 Уровень подготовки администратора

К администраторам Подсистемы предъявляются следующие требования:

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

1.4 Перечень эксплуатационной документации, с которым необходимо ознакомиться администратору

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

2 Общие сведения

Сервис "Запись к врачу" предназначен для создания записи на приём к врачу (без направления), обеспечивает взаимодействия Системы с Голосовым помощником для выполнения записи на прием к врачу.

2.1 Описание информационного взаимодействия

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

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

2.2 Схема взаимодействия

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

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

2.3 Кэширование запросов

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

2.4 HTTP ошибки

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

Общие HTTP ошибки возвращаются HTTP кодами.

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

2.4.1 Ошибки на стороне клиента

Ошибки на стороне клиента представлены в таблице 1.

Таблица 1 – Ошибки на стороне клиента

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

2.4.2 Ошибки на стороне сервера

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

Имеется возможность получения детальной информации по таким ошибкам.

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

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

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

2.4.3 Детализация ошибок

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

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

3 Авторизация

3.1 Авторизация в PHP API. Авторизация в системе

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

Авторизация зависит от используемых механизмов:

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

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

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

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

3.2 Авторизация в SWAN API. Метод авторизации пользователя СМП

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

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

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

Используемый метод – **GET /login**.

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

Интегрируемые системы	– Система – Внешняя ИС
Инициатор сценария:	Внешняя ИС
Предварительные условия:	Сервис развернут. Интеграция включена.
Иницирующее событие:	Вызов прикладного интеграционного метода со стороны внешней ИС
Критерий успешности выполнения:	Авторизация пользователя выполнена. Последующие вызовы интеграционных методов успешны. В случае возникновения ошибок выполнения методов - ошибки возвращаются в описанном формате
Базовые шаги сценария:	1. Внешняя ИС формирует параметры и направляет в ЕЦП запрос GET /rest-api/login (Метод авторизации пользователя) на авторизацию пользователя Системы. 2. При возникновении ошибок на стороне сервера, при выполнении методов, внешняя ИС формирует параметры и направляет в Систему запрос GET /app/errors (Метод получения детальной информации об ошибке) на получение детальной информации об ошибке

4 Описание сценария сервиса "Запись к врачу"

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

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

	<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 приёма).</p> <p>20. Система возвращает свободные даты приёма.</p> <p>21. ГП направляет в Систему запрос GET api/TimeTableGraf/TimeTableGrafFreeTime (Получение свободного времени приёма).</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>Альтернативные шаги сценария:</p>	<p>ЕНП ОМС не найден в системе</p> <p>5А. Система направляет в ГП пустой ответ.</p> <p>6А. ГП переводит пациента на оператора</p> <hr/> <p>Можно записаться не по месту основного прикрепления</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>У пациента уже существует запись к врачу</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>Пациент хочет записаться к конкретному врачу</p> <p>18А. ГП перечисляет врачей, к которым можно записаться.</p> <p>19А. Переход к шагу 19 основного сценария</p> <hr/> <p>Пациенту не подходит ни одна из существующих бирок</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> <p>25А. Система направляет в ГП данные о записи в лист ожидания</p>

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

5.4.1 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 5 – Параметры метода GET api/PersonList

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

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

5.4.2 Получение информации по человеку GET api/Person

Метод предназначен для получения информации по пациенту.

Таблица 6 – Параметры метода GET api/Person

Входящие параметры	<ul style="list-style-type: none"> – Person_id (N, H) – идентификатор пациента; – PersonSurName_SurName (T[50], H) – фамилия; – PersonFirName_FirName (T[50], H) – имя; – PersonSecName_SecName (T[50], H) – отчество; – PersonBirthDay_BirthDay (D, H) – дата рождения; – PersonSnils_Snils (N[11], H) – СНИЛС; – Polis_Ser (T[10], H) – серия полиса; – Polis_Num (N[15], H) – номер полиса; – Lpu_id (N,H) – идентификатор медицинской организации (МО). Если параметр не указан, то метод вернёт данные для всех медицинских организаций текущего региона; – KLCountry_id (N,H) – идентификатор страны гражданства; – Document_IsTwoNation (N,H) – признак "Двойное гражданство"; параметр принимает значения "0" – нет, "1" – да; – allFields (boolean, H) – все поля. Если передан параметр allFields или ResponseFull, то в ответ выводятся вся информация по человеку; – ResponseFull (boolean, H) – полный ответ. Если передан параметр allFields или ResponseFull, то в ответ выводятся вся информация по человеку; – OtherRegion_MoAttach (boolean, H) – пациенты, прикрепленные к МО другого региона
Ответ	<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"> – Если не передан ни один параметр, то ошибка:

- код состояния: 200
- значение параметров:
 - error_code:6
 - error_msg:"В запросе должен быть хотя бы один параметр."
- Если не найдено ни одной записи:
 - код состояния: 200
 - значение параметров:
 - error_code: 0
 - data: []
- Если при обработке запроса обнаружено, что в параметре Polis_Num введено значение, не удовлетворяющее контролю контрольного разряда:
 - код состояния: 200;
 - значение параметров:
 - error_code:6;
 - error_msg:"Номер ЕНП заполнен не верно".
- Если хотя бы одна запись найдена, то успешный ответ – список людей с информацией:
 - Person_id (N, O) – идентификатор пациента;
 - PersonSurName_SurName (T[50], O) – фамилия;
 - PersonFirName_FirName (T[50], H) – имя;
 - PersonSecName_SecName (T[50], H) – отчество;
 - PersonBirthDay_BirthDay (D, H) – дата рождения;
 - Person_Sex_id (N, H) – пол;
 - PersonPhone_Phone (T, H) – телефон;
 - PersonSnils_Snils (N[11],H) – СНИЛС;
 - PersonInn_Inn (N, H) – ИНН пациента;
 - SocStatus_id (N, H) – социальный статус;
 - SocStatusFed_code (T[20]) – социальный статус по коду справочника Минздрава;
 - Person_pid (N, H) – идентификатор представителя пациента;
 - ParPersonSecName_SecName (T[50]) – отчество представителя пациента;
 - DeputyKind_id (N, H) – статус представителя;
 - UAddress_id (N, H) – адрес регистрации;
 - UAddress_Address (T[300]) – адрес регистрации строкой;
 - PAddress_id (N, H) – адрес проживания;
 - PAddress_Address (T[300]) – адрес проживания строкой;
 - Org_id (N, H) – место работы;
 - BAddress_id (N, H) – адрес рождения;
 - Org_id (N, H) – место работы;
 - Post_id (N, H) – должность;
 - BDZ_guid (N, H) – GUID по БДЗ;
 - BDZ_id (N, H) – идентификатор по БДЗ;
 - LegalStatusVZN_id (N) – идентификатор правового статуса нерезидента
 - LegalStatusVZN_Name (T) – наименование правового статуса нерезидента;
 - LegalStatusVZN_pid (N) – идентификатор базового

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

5.4.3 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 7 – Параметры метода **POST api/Person**

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

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

	<ul style="list-style-type: none"> - код статуса: 200 - значение параметров: <ul style="list-style-type: none"> - error_code: 6 - error_msg: "Отсутствует один из обязательных параметров PersonSurName_SurName, PersonBirthDay_BirthDay, Person_Sex_id, SocStatus_id" <p>Если в параметре SocStatus_id указано значение "99", то сохранение значения социального статуса осуществляется в зависимости от возраста человека на текущую дату (возраст человека определяется по значению во входном параметре PersonBirthDay_BirthDay запроса:</p> <ul style="list-style-type: none"> - <=7; - >= возраст на текущую дату <= 18; - >18 возраст на текущую дату <=60; - возраст на текущую дату > 60;
<p>Описание ошибок</p>	<ul style="list-style-type: none"> - признак "Личность неизвестна" (Person_isUnknown) не заполнен или его значение "Нет", и заполнены не все обязательные параметры – отображается ошибка "Отсутствует один из обязательных параметров PersonSurName_SurName, PersonBirthDay_BirthDay, Person_Sex_id, SocStatus_id"; - значение признака "Личность неизвестна" (Person_isUnknown) "Да" и заполнены не все обязательные параметры – отображается ошибка "Отсутствует параметр PersonSurName_SurName"; - в БД есть человек с такими же параметрами, которые указаны во входных параметрах метода – отображается ошибка "Человек с такими данными уже присутствует в базе"; - параметр PersonPhone_Phone содержит не десятизначное число – отображается ошибка "Неверный формат телефонного номера. Телефонный номер должен состоять из 10 цифр"; - в любом передаваемом параметре передается значение, не соответствующее ожидаемому формату – отображается ошибка "Неверный формат поля <Наименования полей через запятую>"; - ожидается параметр, состоящий из 11 символов, а передан параметр, состоящий из большего или меньшего количества символов – отображается ошибка "Номер СНИЛС должен содержать 11 символов"; - переданная дата рождения больше текущей даты – отображается ошибка "Некорректное значение в поле PersonBirthDay_BirthDay"; - в передаваемом параметре (Person_Sex_id, SocStatus_id, Address_id, PAddress_id, BAddress_id, Org_id, Post_id) передается значение id, не найденное в Системе – отображается ошибка "Значения для <Наименования полей через запятую> не найдены в справочниках Системы"; - значение параметров, по которым ожидается значение 0/1 (да/нет) (Person_isUnknown), не найденное в Системе –

	<p>отображается ошибка "Значение поля <Названия полей через запятую> не соответствует ожидаемому";</p> <ul style="list-style-type: none"> – в общих настройках параметров Системы в Контроле на обязательность полей для "Контроля на корректность ИНН" установлено значение "Запрет сохранения", и в поле ИНН было передано непустое значение, проводится проверка на контрольную сумму – отображается ошибка "Ошибка проверки контрольной суммы в ИНН"; – включена проверка контрольной суммы СНИЛС, и она не пройдена – отображается ошибка "Ошибка проверки контрольной суммы в СНИЛС"; – включена проверка СНИЛС на дублирование, и она не пройдена – отображается ошибка "Человек с введённым номером СНИЛС уже есть в базе"
--	---

5.4.4 Редактирование данных человека PUT api/Person

Метод предназначен для редактирования информации по человеку.

Таблица 8 – Параметры метода PUT api/Person

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

- пациента;
- PersonFamilyStatus_IsMarried (N,H) – идентификатор типа нахождения в зарегистрированном браке;
 - DocumentAuthority_id (N,H) – идентификатор типа документа законно представителя;
 - DocumentDeputy_Ser (T, H) – серия документа законного представителя;
 - DocumentDeputy_Num (T, H) – номер документа законного представителя;
 - DocumentDeputy_Issue (T, H) – кем выдан документ законного представителя;
 - DocumentDeputy_begDate (D,H) – дата выдачи документа законного представителя;
 - DeputyOrg_id (N,H) – идентификатор организации-попечителя;
 - CheckDeputyTies (N,H) – признак: контролировать допустимость типа представителя (Если контроль включён и тип недопустим, то метод вернёт ошибку);
 - options (S, H) – настройка работы метода. Допустимые форматы: JSON или массива. Параметры настройки:
 - CheckChecksumSnilsOff (N, H):
 - 1 – отключить проверку контрольной суммы СНИЛС;
 - 0 или пропущен – проверять контрольную сумму СНИЛС;
 - CheckDuplicateSnilsOff (N,H):
 - 1 – отключить проверку дублирования СНИЛС;
 - 0 или пропущен – проверять СНИЛС на дублирование;
 - DuplicateSearch (T,H) – список дополнительных параметров для поиска "двойников". Допустимы форматы: JSON, список. Допустимые параметры: PersonSnils_Snils, PersonInn_Inn, Polis_EdNum.

При передаче пустых параметров можно очистить все поля пациента, кроме PersonSurName_SurName, PersonBirthDay_BirthDay, Person_Sex_id, SocStatus_id.

Текущие правила:

- Нельзя указать одновременно Person_pid и DeputyOrg_id. Один из них должен быть пустым или не указан.
- Если параметр DeputyKind_id указан и пустой – запись об организации-попечителе удаляется.
- Если параметр DeputyKind_id указан и =1 (тип "Отсутствует") – информация об организации-попечителе удаляется. При этом параметр DeputyOrg_id игнорируется.
- Если указан только DeputyOrg_id и он пустой – текущая запись об организации-попечителе удаляется.
- Если указан DeputyOrg_id и он не пустой – в текущей записи о представителе записывается информация об организации-попечителе. Одновременно удаляется информация о человеке-представителе (согласно правилу 1).

	<ul style="list-style-type: none"> - Если представитель добавляется (т.е. представителя ещё нет у пациента) –параметр DeputyKind_id становится обязательным
<p>Ответ</p>	<ul style="list-style-type: none"> - Проводится проверка: параметр PersonPhone_Phone может содержать только 10–значное число. Если результат проверки отрицательный, то выводится ошибка о неверном формате входного параметра; - Если по переданному значению Person_id не найден пациент, то ошибка "Пациент не найден в системе". - Если в любом передаваемом параметре передается значение, не соответствующее ожидаемому формату, например: <ul style="list-style-type: none"> - ожидается числовое значение, а передано текстовое значение; - ожидается параметр, состоящий из 11 символов, а передан параметр, состоящий из большего или меньшего количества символов; <p>то ошибка: "Неверный формат поля <Наименования полей через запятую>". Обновление человека в БД не происходит.</p> <ul style="list-style-type: none"> - Проводится проверка на дублирование пациента: если в БД есть человек с такими же параметрами, которые указаны во входных параметрах метода (т.е. если для имеющегося человека Person_id передана Фамилия, по которой вместе с другими обязательными параметрами проверки на дубль будет найден другой Человек с совпадающими параметрами), то ошибка "Данные человека не прошли проверку на дублирование". Данные по человеку в БД не обновляются. - Если в любом передаваемом параметре передается значение id (Person_Sex_id, SocStatus_id, Address_id, PAddress_id, VAddress_id, Org_id, Post_id), не найденное в системе, ошибка: "Значения для <Наименования полей через запятую> не найдены в справочниках системы", обновление данных человека в БД не происходит. Аналогичный контроль производится для параметров, по которым ожидается значение 0/1 (да/нет) (Person_isUnknown). Если контроль не пройден, то ошибка "Значение поля <названия полей через запятую> не соответствует ожидаемому", обновление данных человека в БД не происходит. - Производится проверка: параметр СНИЛС может содержать только 11–тизначное значение (при проверке исключаются дефисы, которые могут быть переданы в параметре SocStatus_id), состоящее из цифр. Если результат проверки отрицательный, то выводится ошибка о неверном формате входного параметра. - Правила для параметров: <ul style="list-style-type: none"> - PersonSurName_SurName (Т, У): ... <p>Если параметр не указан, то он не проверяется.</p> <p>Если параметр указан и пустой, то метод вернёт ошибку: Значения для</p>

(PersonSurName_SurName) обязательное и не может быть пустым.

- PersonBirthDay_BirthDay (D, Y): ...

Если параметр не указан, то он не проверяется.

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

- Person_Sex_id (N, Y): ...

Если параметр не указан, то он не проверяется.

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

- SocStatus_id (N, Y): ...

Если параметр не указан, то он не проверяется.

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

Если запрос обработан корректно, и данные успешно обновлены, то:

- код состояния: 200;
- значение параметров:
- success: true;
- error_code: 0;
- data:
 - Person_id: значение параметра.

Если в запросе не указан обязательный параметр Person_id, то:

- код состояния: 200;
- значение параметров:
- error_code: 6;
- error_msg: Отсутствует обязательный параметр 'Person_id'.

Если в запросе указан неверный формат номера:

- код состояния: 200;
- значение параметров:
- success: false;
- error_code: 6;
- error_msg: Неверный формат телефонного номера. Телефонный номер должен состоять из 10 цифр

5.4.5 Получение записей на прием к врачу GET `api/TimeTableGraf/TimeTableGrafByMedStaffFact`

Таблица 9 – Параметры метода GET `api/TimeTableGraf/TimeTableGrafByMedStaffFact`

Входящие параметры	<ul style="list-style-type: none"> – MedStaffFact_id (N, O) – место работы врача – TimeTableGraf_beg (DT, O) – дата и время начала диапазона – TimeTableGraf_end (DT, O) – дата и время окончания диапазона
Ответ	<p>Успешный ответ – список бирок и пациентов, записанных к врачу в указанный диапазон:</p> <ul style="list-style-type: none"> – TimeTableGraf_id (N, O) – идентификатор бирки – TimeTableGraf_begTime (DT, O) – дата и время приема – TimeTableGraf_factTime (DT, H) – фактическое время приема. – Person_id (N, O) – идентификатор пациента

5.4.6 Получение прикреплений пациента GET `api/PersonAttach`

Таблица 10 – Параметры метода GET `api/TimeTableGraf/TimeTableGrafByMedStaffFact`

Входящие параметры	<ul style="list-style-type: none"> – Person_id (N, O) – идентификатор пациента; – Lpu_id (N, H) – идентификатор МО; – LpuRegion_id (N, H) – идентификатор участка; – LpuAttachType_id (N, H) – идентификатор типа прикрепления
Ответ	<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"> – PersonSnils_Snils (N[11], H) – СНИЛС; – Org_id (N, H) – место работы; – Post_id (N, H) – должность. <p>Список прикреплений:</p> <ul style="list-style-type: none"> – PersonCard_id (N, H) – идентификатор прикрепления; – PersonCard_begDate (D, H) – дата прикрепления; – PersonCard_endDate (D, H) – дата открепления; – PersonCard_Code (N, H) – номер амбулаторной карты; – Lpu_id – идентификатор МО прикрепления; – Lpu_Name – полное наименование МО прикрепления; – Lpu_Nick – краткое наименование МО прикрепления; – LpuRegion_Name – наименование участка; – LpuRegion_id (N, H) – идентификатор участка прикрепления;

	<ul style="list-style-type: none"> – LpuAttachType_id (N, H) – идентификатор типа прикрепления; – PersonCard_isAttachAuto (N, H) – признак автоприкрепления; – CardCloseCause_id (N, H) – идентификатор причины закрытия карты; – PersonCard_AttachAutoDT (DT, H) – дата и время автоматического прикрепления; – PersonCard_isAttachCondit (N, H) – признак условного прикрепления
Пример	<pre>{ "error_code": 0, "data": [{ "PersonSnils_Snils": null, "Org_id": "1190018853", "Post_id": null, "attach_data": [{ "PersonCard_id": "77306518970", "PersonCard_begDate": "2020-04-02", "PersonCard_endDate": null, "PersonCard_Code": "943218325458", "Lpu_id": "10010833", "Lpu_Name": "ГБУЗ П КРАЯ ГОРОДСКАЯ КЛИНИЧЕСКАЯ БОЛЬНИЦА №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>

5.4.7 Получение основного прикрепления пациента GET api/PersonMainAttach

Таблица 11 – Параметры метода GET api/PersonMainAttach

Назначение метода	Возвращаются данные по МО действующего основного прикрепления пациента (тип участка – терапевтический или педиатрический, если пациенту меньше 18
--------------------------	---

	лет), а также данные участкового врача.
Входящие параметры	Person_id (N, O) – идентификатор пациента.
Ответ	<p>Успешный ответ:</p> <ul style="list-style-type: none"> – Person_id (N, O) – идентификатор пациента; – Lpu_id (N, O) – идентификатор МО прикрепления; – LpuUnit_FRMOid (T, O) – единый уникальный идентификатор медицинской организации (OID). Порядок определение OID: – У группы отделений заполнено поле OID ФРМО (LpuUnit.FRMOid); – Иначе, у группы отделений сохранена ссылка на справочник "ФРМО. Справочник структурных подразделений"; – Отделение из группы отделений связано с записью справочника "ФРМО. Справочник отделений и кабинетов": по ней определяется связанная запись по справочнику "ФРМО. Справочник структурных подразделений"; – В случае отсутствия OID у структурного подразделения передается OID МО (используется PassportToken_tid из fed.PassportToken, передаются только OID нового образца); – Lpu_Name (T, O) – наименование МО. Порядок определения наименования: < Наименование МО> +<Наименование Группы отделений>, где Наименование Группы отделений определяется след. образом: – Если у группы отделений заполнено поле OID ФРМО (LpuUnit.FRMOid), то наименование группы отделений; – Наименование медицинской организации; – Наименование МО в Реестре МО РФ (fed.PassportToken); – Address_Address (T, O) – адрес фактического местоположения структурного подразделения медицинской организации. Указывается адрес структурного подразделения медицинской организации следующем формате: почтовый индекс, регион, населенный пункт, улица, дом. Пример: 411228, Саратовская обл., г. Саратов, ул. им. Черемушкина Н. Г., д. 141; – Org_Phone (T, 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;

	<ul style="list-style-type: none"> – проверить признак основного врача на участке (MedStaffRegion_isMain = 2); – если на участке нет основного врача, в период которого попадает дата запроса, то выбрать первого из списка врача, у которого текущая дата запроса попадает в период работы на участке; – MedStaffFact_Fio (Т, О) – Ф. И. О. врача участка
Ошибки	<p>Возможные ошибки:</p> <ul style="list-style-type: none"> – если в запросе не заполнен входящий параметр, отобразится ошибка (прикладной код: "6") с текстом: "Отсутствует обязательный параметр 'Person_id'"; – если по Person_id не найдено действующего основного прикрепления пациента, отобразится ошибка (прикладной код: "1") с текстом: "Не найдено действующего основного прикрепления пациента!"
Пример	<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 12 – Параметры метода GET api/Refbook

Входящие параметры	<ul style="list-style-type: none">– Refbook_Code – Код справочника;– Refbook_TableName – Наименование таблицы справочника;– Code – Код элемента справочника;– Id – идентификатор элемента справочника;– Name – Наименование элемента справочника– Offset – Номер страницы;– Limit – Максимальное число возвращаемых записей;– Page – номер начальной возвращаемой записи
Пример запроса	GET {{url}}/api/Refbook?Refbook_TableName=dbo.socstatus&Page=2&limit=2
Ответ	<ul style="list-style-type: none">– data:– id – Идентификатор элемента справочника;– OrgType_id – Идентификатор типа организации;– Name – Наименование элемента справочника;– Code – Код элемента справочника;– KLRgn_id – Идентификатор региона;– begDate – Дата начала;– endDate – Дата окончания
Пример ответа	<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> >Name": "Военнослужащий МВД", "Code": "4", "KLRgn_id": null, "begDate": null, "endDate": "2020-05-01" }] } </pre>
<p>Ошибки</p>	<ul style="list-style-type: none"> - Проверить наличие обязательных параметров: <ul style="list-style-type: none"> - 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. - Иначе перейти к следующему шагу. - Провалидировать значения входящих параметров. Если нарушены правила валидации, то сформировать ошибку из списка ниже. <ul style="list-style-type: none"> - Если значение параметра, являющегося идентификатором, не является целым числом больше нуля, то сформировать ошибку 6 Ошибка формата идентификатора. Перейти к шагу 4. - Если переданные значения не соответствуют заявленным типам, то сформировать ошибку 6 Ошибка соответствия параметров заявленным типам. Перейти к шагу 4. - Проверить данные справочников: <ul style="list-style-type: none"> - 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. - Если найдено больше одного справочника подходящего под условия запроса, то сформировать ошибку 6 По указанному коду найдено несколько справочников. Перейти к шагу 4. - Если поле отсутствует в запрашиваемом справочнике, то сформировать ошибку 6 Поля нет в справочнике. Перейти к шагу 4.

Алгоритм завершает работу

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

Мобильный метод предназначен для поиска человека по атрибутам.

Таблица 13 – Параметры метода GET /api/Person/mgetPersonSearch

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

	<ul style="list-style-type: none"> - attbefore3 - только прикрепленные или старше 3 лет - att_vol - только прикрепленные и в объеме - att_1_4 - только прикрепленные (основное или служебное) - att - только прикрепленные - person_register_by_vzn_request - поиск в регистре для заявки ВЗН - hms - поиск для регистра главных внештатных специалистов - fmba - только не включенные в регистр ФМБА - dd - если ищем по ДД - encryonly - отображение только зашифрованных ВИЧ-инфицированных - men_only - women_only - только эм или жо - smo или smo3 - Поиск в арм смо. Для Перми - последние 3 года - erssnils - снилс должен быть заполнен
<p>Пример</p>	<p>Успешный ответ:</p> <pre> { "error_code": 0, "data": [{ "Person_id": "100126424", "Server_id": "3", "PersonEvn_id": "88401919", "Person_IsInErz": null, "Person_Phone": null, "Person_Inn": null, "Person_IsUnknownSign": null, "Person_isOftenCaller": 1, "Person_IsRefuse": "true", "Person_IsDead": "false", "Person_IsFedLgot": "false", "Person_IsRegLgot": "false", "Person_Lgots": "false", "Person_Is7Noz": "false", "UAddress_AddressText": null, "PAddress_AddressText": null, "Person_IsBDZ": null, "PersonQuarantine_begDT": null, "PersonQuarantine_IsOn": "false", </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>
Ответ	<p>Логика формирования параметров ответа:</p> <p>Person_IsBDZ. Значение по умолчанию:</p> <ul style="list-style-type: none"> - Если Polis_endDate не пустое поле и Polis_endDate меньше текущей даты, то 'orange'. - Если person.PersonCloseCause_id= 2 и person.Person_closeDT не пустое поле, то 'red' - Если person.server_id = 0, то 'true'Иначе 'false' - Для Перми: <ul style="list-style-type: none"> - Если person.server_id = 0, то 'true' - Если person.Person_IsInErz = 1, то 'blue' - Если person.server_id = 0 и Polis_endDate меньше текущей даты, то - Если Person_deadDT - пустое поле, то 'yellow' Иначе - 'red' - Если person.server_id = 2 и person.Person_IsInErz не равно 1, то 'false' - Для Пензы: <ul style="list-style-type: none"> - Если person.Person_IsInErz = 1, то 'orange' - Если person.Person_IsInErz = 2, то 'true' - Иначе 'false' - Для Казахстана - <ul style="list-style-type: none"> - Если person.Person_IsInErz = 1, то 'red' - Если person.Person_IsInErz = 2, то 'true'Иначе 'false'
Ошибки	<p>Если не указаны параметры запроса:</p> <pre> { "success": false, "error_msg": "Не указан ни один фильтр. Для поиска заполните хотя бы один параметр в блоке фильтров.", "error_code": 6 } </pre>

5.6 Методы работы с листами ожидания

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

Таблица 14 – Параметры метода POST api/EvnQueue/EvnQueue

Входящие параметры	<ul style="list-style-type: none">– Person_id (N, O) – идентификатор пациента;– LpuSectionProfile_id (N, Y) – идентификатор профиля отделения МО.– LpuSection_id (N, H) – идентификатор отделения.– MedPersonal_id (N, H) – направивший врач;– MedStaffFact_did (N, Y) – врач, к кому в лист ожидания включается пациент. Указывается, если запись выполняется к конкретному врачу.– Lpu_sid (N, H) – направившее МО;– Lpu_did (N, O) – МО, куда направили;– EvnQueue_desDT (D, H) – дата и время посещения (желаемая);– MedService_did (N, Y) – идентификатор службы, в которую нужно поставить в очередь;– UslugaComplex_did (N, O) – идентификатор услуги, на которую необходимо записать
Действия при получении запроса	<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">– ожидается числовое значение, а передано текстовое значение;– ожидается параметр, состоящий из 11 символов, а передан параметр, состоящий из большего или меньшего количества символов; <p>то, ошибка: "Неверный формат поля <Наименования полей через запятую>", 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</p>

	<p>системе, ошибка: "Значения для <Наименования полей через запятую> не найдены в справочниках системы". Создание записи в БД не происходит. Если есть ошибки, то возвращается ошибка. Иначе:</p> <ul style="list-style-type: none"> – в лист ожидания добавляется запись – пациент ставится в очередь по профилю или врачу (в соответствии с полученными параметрами). – статус записи "в очереди" (EvnQueueStatus_id). – способ записи/включения в Лист ожидания – "РИШ"
Ответ	<p>Если есть ошибки, то возвращается код ошибки. Иначе:</p> <ul style="list-style-type: none"> – EvnQueue_id (N, O) – идентификатор постановки в очередь – EvnQueueStatus_id (N, H) – статус листа ожидания

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 15 – Параметры метода **GET api/PostByName**

Входящие параметры	<ul style="list-style-type: none"> – post_name (T, O) – наименование должности; – sess_id (N, O) – идентификатор сессии
Ответ	<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"> – error_code: 0 – data: – Post_id: – Post_name: <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/MOById

Таблица 16 – Параметры метода **GET api/MOById**

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

5.7.3 Получение информации о докторе GET /api/doctors/MedStaffFact_id

Метод позволяет получить информацию о враче.

Таблица 17 – Параметры метода **GET /api/doctors/MedStaffFact_id**

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

	<ul style="list-style-type: none"> - annot (T) – примечание; - status (N) – статус. Возможные значения: - 1 – есть ближайшая свободная бирка; - 2 – нет расписания; - status_msg (T) – текст с первой ближайшей записью; - can_post_feedback (B) – возможность комментирования. Возможные значения: - true – да, - false – нет - doctorInfoList (S, H) – информация о врачах при установленном фильтре doctorInfoList: - MedStaffFact_id (S) – идентификатор места работы врача; - IsPaidRec (N) – платный врач; - MedPersonal_FullName (S) – Ф. И. О. врача; - Lpu_Nick (S) – название МО; - LpuUnit_Address (S) – адрес МО; - QualificationCat_Name (S) – квалификация врача; - feedback_total (N) – количество оценок; - feedback_rating (N) – рейтинг врача; - firstFreeDate (S) – дата ближайшей бирки; - ProfileSpec_Name (S) – специальность; - videochatIsOn (B) – флаг - видеочат включен ; - IsAttachedDoctor (B) – находится ли доктор в прикреплении у пациента; - rating_avg (N) – рейтинг врача; - avatar_path (S) – аватар врача; - lng (N) – координаты долготы; - lat (N) – координаты широты; - firstFreeDates (S) – дата/время бирки; - source_system (S) – источник поиска врачей: pm_main – основные, pm_paid – платные
Ответ	<p>Успешный ответ:</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 18 – Параметры метода GET api/MedSpecOms/MedSpecOmsByMO

Входящие параметры	<ul style="list-style-type: none"> – Lpu_id (N, O) – Идентификатор МО – For_Record (N, Y) – параметр для определения первой свободной даты записи. Если передан, то будет считаться количество бирок доступных для записи с типами: Обычная, Запись через инфомат, Для интернета.
Ответ	<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"> – MedSpecOms_id (N, O) – Идентификатор специальности. – TimetableGraf_Count (N, Y) – Количество доступных бирок для записи (выгружается, если For_Record = 1)

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 19 – Параметры метода **GET api/MedStaffFact/MedStaffFactByMO**

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

	<p>врача. Если параметр TimeTableType_id имеет пустое значение или значение в параметре не указано, то в параметре TimetableGraf_Count будет указано количество свободных биров с любым типом</p>
<p>Пример</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", </pre>

	<pre> "PersonFirName_FirName": "ТЕСТ", "PersonSecName_SecName": "", "RecType_id": null, "LpuUnitType_id": "9", "MedStaffFact_setDate": "01.01.2021", "MedStaffFact_disDate": null, "LpuBuilding_id": "10101000000074" }] } </pre>
--	--

5.7.6 Получение свободных дат приема GET api/TimeTableGraf/TimeTableGrafFreeDate

Таблица 20 – Параметры метода GET api/TimeTableGraf/TimeTableGrafFreeDate

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

5.7.7 Получение свободного времени приема GET api/TimeTableGraf/TimeTableGrafFreeTime

Таблица 21 – Параметры метода GET api/TimeTableGraf/TimeTableGrafFreeTime

Входящие параметры	<ul style="list-style-type: none"> – MedStaffFact_id (N, O) – место работы врача; – TimeTableGraf_begTime (D, O) – свободная дата приема; – Person_id (N, H) – идентификатор пациента. Правила: – если Person_id не указан, то возвращаются все виды бирок,
---------------------------	---

	<p>кроме вида "Диспансерный учёт";</p> <ul style="list-style-type: none"> – если Person_id указан, и у пациента нет открытой карты диспансерного наблюдения, то возвращаются все виды бирок, кроме вида "Диспансерный учёт"; – если Person_id указан и у пациента есть открытая карта диспансерного наблюдения, то возвращаются все виды бирок, в том числе вида "Диспансерный учёт"; – TimeTableType_id (N, H) – тип бирки; – TimeTableGrafFreeDate (N, H) – параметр учета прошедшего времени ("1" – да, "0" – нет). Если значение не указано или указано "0", то метод возвращает бирки со временем начала, большим текущего времени. Если в параметре передано значение "1", то метод возвращает все бирки за указанную дату; – offset (N, H) – номер начальной возвращаемой записи; – limit (N, H) – максимальное число возвращаемых записей. Значение по умолчанию – 10000; – page (N, H) – номер страницы. Значение по умолчанию – 1
Ответ	<p>Успешный ответ – массив свободных бирок на прием к врачу: {</p> <ul style="list-style-type: none"> – TimeTableGraf_id (N, H) – идентификатор свободной бирки. Если по пациенту (Person_id) найдена открытая контрольная карта диспансерного наблюдения в отделении МО места работы врача и специальность врача совпадает со специальностью ответственного врача, указанного в карте дисп. наблюдения, то дополнительно передаются идентификаторы свободных бирок с типом "Диспансерный учёт"; – TimeTableGraf_begTime (DT, O) – дата и время начала приема; – TimeTableGraf_Time (N,O) – длительность приема. <p>}</p> <p>Выходной параметр для случаев, когда бирка, на которую пытаются записаться, заблокирована:</p> <ul style="list-style-type: none"> – TimeTableStatusLockComment (T, H) – сообщение о блокировке бирки. Сообщение содержит в себе возможное время разблокировки бирки. <p>Проверка на наличие переданной во входных параметрах бирки в таблице БД по TimeTableMedService_id:</p> <ul style="list-style-type: none"> – если TimeTableMedService_id есть в БД, записывать на бирку

	<p>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;</p> <ul style="list-style-type: none"> – в связи с данной проверкой для бирок, которые нашлись в таблице БД выводится в ответе параметр: – TimeTableStatusLockComment, в котором будет содержаться следующая информация: "На данный момент бирка редактируется оператором и может быть занята. Время возможного высвобождения бирки и доступности записи на нее 2017-12-23 21:59:34.260000" (т.е. в конце комментария выводится время из TimeTableLock_lockTime + 15 мин)
--	--

5.7.8 Запись пациента на прием POST api/TimeTableGraf/TimeTableGrafWrite

Таблица 22 – Параметры метода POST api/TimeTableGraf/TimeTableGrafWrite

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

	<p>время разблокировки бирки.</p> <p>Проверка на наличие переданной во входных параметрах бирки в таблице БД по TimeTableMedService_id:</p> <ul style="list-style-type: none"> – если TimeTableMedService_id есть в БД, записывать на бирку нельзя. Производится подсчет минут от даты и времени, указанных в БД исходя из логики, что бирка может быть заблокирована оператором 15 минут. Соответственно, если TimeTableLock_lockTime=2017-12-23 21:59:34.260000, то бирка станет доступной для записи или занятой в 2017-12-23 22:14:34.260000; – в связи с данной проверкой для бирок, которые нашлись в таблице БД выводится в ответе параметр: – TimeTableStatusLockComment, в котором будет содержаться следующая информация: "На данный момент бирка редактируется оператором и может быть занята. Время возможного высвобождения бирки и доступности записи на нее 2017-12-23 21:59:34.260000" (т.е. в конце комментария выводится время из TimeTableLock_lockTime + 15 мин). <p>Успешный ответ – составной идентификатор записи на прием:</p> <ul style="list-style-type: none"> – Person_id (N, O), TimeTableGraf_id (N, O) – составной идентификатор записи на прием. <p>Если во входящих параметрах указан идентификатор постановки в очередь, то в направление передается TimeTableGraf_id, а Evn_Queue_id удаляется.</p>
--	--

5.7.9 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 23 – Параметры метода **GET api/TimeTableList**

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

	<p>передано значение "MedService", то метод вернёт бирки в соответствии с указанным типом (на службу), переданные значения параметра MedStaffFact_id будут проигнорированы;</p> <ul style="list-style-type: none"> – если передано значение и при этом параметр TimeTableSource не передан, то метод вернёт все бирки с типом "на службу" и те бирки с типом "к врачу", которые соответствуют переданным значениям параметра MedStaffFact_id; – Post_id (N, H) – идентификатор должности. Правила передачи: – если передано значение и в параметре TimeTableSource передано значение "MedService", то метод вернёт бирки в соответствии с указанным типом (на службу), переданные значения параметра Post_id будут проигнорированы; – если передано значение и при этом TimeTableSource не передан, то метод вернёт все бирки с типом "на службу" и те бирки с типом "к врачу", которые соответствуют переданным в параметре Post_id. – TimeTable_end (DT,H) – дата и время окончания диапазона; – TimeTableSource (T, H) – тип бирки. Возможные значения: Graf/MedService для получения бирок на прием к врачу/на услугу службы соответственно; – offset (N,H,0) – номер начальной возвращаемой записи; – limit (N,H,100) – максимальное число возвращаемых записей; – Page (N,H,1) – номер страницы.
<p>Ответ</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"> – Lpu_id (N, O) – идентификатор МО; – Org_Name (T, O) – Наименование МО; – TimeTables(S, O) – объект; – Person_id (N, O) – идентификатор пациента; – Person_SurName (T, O) – фамилия пациента;

	<ul style="list-style-type: none"> – Person_FirName (Т, О) – имя пациента; – Person_SecName (Т, О) – отчество пациента; – Person_Phone (Т, О) – номер телефона пациента; – TimeTableSource (Т, О) – тип бирки:Graf или MedService; – TimeTable_id (N, O) – идентификатор бирки; – TimeTableType_id (N, O) – тип бирки; – TimeTable_factTime (DT, H) – фактические дата и время приема для обслуженных бирок (если у бирки есть Фактическое время приема TimeTableGraf_factTime/TimeTableMedService_factTime, то пациент приходил); – TimeTable_begTime (DT, O) – дата и время начала приёма; – MedPersonal_id (N) – идентификатор сотрудника; – MedPersonal_SurName (Т) – фамилия сотрудника; – MedPersonal_FirName (Т) – имя сотрудника; – MedPersonal_SecName (Т) – отчество сотрудника; – MedPersonal_FIO (Т) – Ф. И. О. сотрудника в нормализованном виде. – MedService_id (N, H) – идентификатор службы (выводить только если тип бирки MedService); – MedService_Name (Т, H) – наименование службы (выводить только если тип бирки MedService); – MedStaffFact_id (N, H) – идентификатор врача (выводить только если бирка поликлиническая Graf); – Post_id (N, H) – идентификатор должности врача (выводить только если бирка поликлиническая Graf); – Post_name (Т, H) – наименование должности врача (выводить только если бирка поликлиническая Graf); – LpuSectionProfile_id (N, O) – идентификатор профиля отделения; – Usluga_Code (Т, H) – код услуги; – Usluga_Name (Т, H) – наименование услуги.
Возможные ошибки	<ul style="list-style-type: none"> – код "1", прикладной код "6", текст ошибки: "Неверное значение TimeTableSource. Должно быть Graf или MedService"

	"; – код "2", прикладной код "6", текст ошибки: "Значение MedStaffFact_id или Post_id не найдено в системе"
--	--

5.7.10 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 24 – Параметры метода **GET api/TimeTableListbyPatient**

Входящие параметры	<ul style="list-style-type: none"> – Person_id (N,O) – идентификатор пациента; – TimeTable_beg (DT,H) – дата и время начала диапазона поиска (если не указано, то считать текущие время и дату за дату начала диапазона); – TimeTable_end (DT,H) – дата и время окончания диапазона; – Lpu_id (N,H) – идентификатор медицинской организации (доступно только для вендора)
Ответ	<p>Правила:</p> <ul style="list-style-type: none"> – значение параметра TimeTable_beg не может быть ранее текущих даты и времени. Иначе метод вернёт ошибку; – если параметр TimeTable_beg не задан, то устанавливается равным текущим дате и времени; – значение параметра TimeTable_end не может быть меньше параметра TimeTable_beg. Иначе метод вернёт ошибку; – если идентификатор человека не найден в Системе, то метод вернёт ошибку. <p>Возвращаемые параметры:</p> <ul style="list-style-type: none"> – Person_id (N, O) – идентификатор пациента; – Person_SurName (T, O) – фамилия пациента; – Person_FirName (T, O) – имя пациента; – Person_SecName (T, O) – отчество пациента; – Person_Phone (T, O) – номер телефона пациента; – TimeTable (S, O) – список ячеек расписания, на которые записан человек: – Lpu_id (N, O) – идентификатор медицинской организации (доступно только для вендора);

	<ul style="list-style-type: none"> – TimeTableSource (Т, О) – метка источника данных; – TimeTable_id (N, O) – идентификатор ячейки расписания; – TimeTable_factTime (DT, Н) – фактически дата и время приема для обслуженных ячеек расписания (если у ячейки есть фактическое время приема TimeTableGraf_factTime/TimeTableMedService_factTime, то пациент приходил); – TimeTable_begTime (Т, О) – дата и время начала приёма; – MedService_id (N, Н) – идентификатор службы; – MedService_Name (Т, Н) – наименование службы; – MedStaffFact_id (N, Н) – идентификатор места работы врача; – Post_id (N, O) – идентификатор должности врача; – Post_name (Т, О) – наименование должности врача; – LpuSectionProfile_id (N, O) – идентификатор профиля отделения; – Usluga_Code (Т, Н) – код услуги; – Usluga_Name (Т, Н) – наименование услуги.
--	---

5.7.11 Изменение статуса записи на прием PUT api/TimeTableGraf/TimeTableGrafStatus

Таблица 25 – Параметры метода PUT api/TimeTableGraf/TimeTableGrafStatus

Входящие параметры	<ul style="list-style-type: none"> – Person_id (N, O), TimeTableGraf_id (N, O) – составной идентификатор записи на прием. – EvnStatus_id (N,O) – идентификатор статуса направления – TimeTableGraf_id (N, O) – ссылка на бирку поликлиники
Ответ	<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>

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 характеризующееся 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.

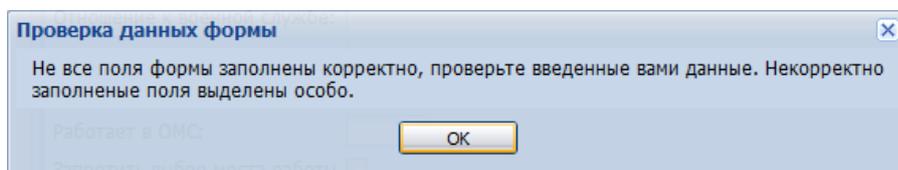
6.2 Действия в случае несоблюдения условий выполнения технологического процесса

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.