

**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 на вакцинацию" 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 функциональных характеристик .....	5
2.2	Описание информационного взаимодействия.....	6
2.3	Схема взаимодействия .....	6
2.4	Кэширование запросов.....	6
2.5	HTTP ошибки.....	6
2.5.1	Ошибки на стороне клиента.....	6
2.5.2	Ошибки на стороне сервера .....	7
2.5.3	Детализация ошибок .....	7
3	Авторизация .....	8
3.1	Авторизация в PHP API. Авторизация в системе.....	8
3.2	Авторизация в SWAN API. Метод авторизации пользователя СМП.....	9
4	Описание сценариев сервиса "Запись на вакцинацию" .....	10
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	Общие методы.....	14
5.4.1	Поиск человека по атрибутам GET /api/Person/mgetPersonSearch.....	14
5.4.2	Получение списка параметров адреса GET api/AddressSearch .....	18
5.5	Методы сервиса "Запись на вакцинацию" .....	22
5.5.1	Получение расписания на вакцинацию в МО прикрепления пациента и ближайшие МО GET api/VaccinationTimeTableMedService.....	22
5.5.2	Запись пациента на приём службы POST api/TimeTableMedService/TimeTableMedServiceWrite .....	23
6	Эксплуатация сервиса.....	25
7	Аварийные ситуации .....	26
7.1	Описание аварийных ситуаций.....	26

7.2 Действия в случае несоблюдения условий выполнения технологического процесса ..27

# **1 Введение**

## **1.1 Область применения**

Настоящий документ описывает порядок работы с сервисом "Запись на вакцинацию" 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 на вакцинацию" 3.0.3\_1 предназначен для записи на вакцинацию с помощью голосового помощника.

## **1.3 Уровень подготовки администратора**

К администраторам Подсистемы предъявляются следующие требования:

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

## **1.4 Перечень эксплуатационной документации, с которым необходимо ознакомиться администратору**

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

## 2 Общие сведения

### 2.1 Описание функциональных характеристик

Сервис "Запись на вакцинацию" 3.0.3\_1 предназначен для записи на вакцинацию с помощью голосового помощника:

- участники процесса:
  - Система:
    - Подсистема "Регистратура";
    - Подсистема "Взаимодействие с внешними голосовыми ботами (в т.ч. технологии ИИ) 122";
    - Подсистема "Иммунопрофилактика".
  - Голосовой помощник (ГП).
- инициирующее событие: Пациент совершает звонок в службу с ГП и выбирает опцию "Запись на вакцинацию";
- критерий успешности выполнения процесса: в ГП передан идентификатор записи на вакцинацию, созданной в Системе;
- базовый сценарий:
  - пациент звонит в службу голосового помощника и выбирает опцию "Запись на вакцинацию";
  - ГП для идентификации пациента просит его назвать единый номер полиса ОМС;
  - пациент называет свои данные;
  - ГП направляет в Систему запрос на поиск человека по атрибутам с названным пациентом ЕНП ОМС;
  - Система отправляет данные о человеке;
  - ГП направляет в Систему запрос на получение списка бирок на вакцинацию в ближайшем к месту прикрепления МО;
  - Система возвращает ГП информацию о бирках в МО прикрепления;
  - ГП называет адрес и уточняет у пациента, будет ли он проходить вакцинацию в МО прикрепления;
  - пациент выбирает дату и время вакцинации;
  - ГП направляет в Систему запрос на запись пациента на приём службы;
  - Система создаёт талон на вакцинацию;
  - Система возвращает идентификатор талона ГП.

## 2.2 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.

## 2.3 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.

## 2.4 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.

## 2.5 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.

### 2.5.1 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.

### 2.5.2 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.

### 2.5.3 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 – GET api/user/login (Авторизация пользователя в системе).</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 сервиса "Запись на вакцинацию"

Таблица 4 – 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 прикрепления.</li> <li>8. ГП называет адрес и уточняет у пациента, будет ли он проходить вакцинацию там</li> <li>9. Пациент выбирает дату и время вакцинации.</li> <li>10. ГП направляет в Систему запрос POST api/TimeTableMedService/TimeTableMedServiceWrite (Запись пациента на приём службы).</li> <li>11. Система создаёт талон на вакцинацию.</li> <li>12. Система возвращает в ответе метода идентификатор талона</li> </ol>
<b>Альтернативные шаги сценария:</b>	<p><b>ЕНП ОМС не найден в системе</b></p> <ol style="list-style-type: none"> <li>5А. Система направляет в ГП пустой ответ.</li> <li>6А. ГП переводит пациента на оператора</li> </ol> <p><b>Пациент хочет пройти вакцинацию по другому адресу</b></p> <ol style="list-style-type: none"> <li>9А. ГП узнаёт у пациента данные о его адресе</li> <li>10А. ГП направляет в ЕЦП запрос GET api/AddressSearch (Получение списка параметров адреса).</li> <li>11А. Система проверяет наличие адреса в системе.</li> <li>12А1. Переход к шагу 9 базового сценария</li> </ol> <p><b>Адрес не найден в системе</b></p>

	12В. Система возвращает пустой ответ. 13В. ГП переводит пациента на оператора
--	--

## 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 Общие методы

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

Мобильный метод предназначен для поиска человека по атрибутам.

Таблица 5 – 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>– hasopenevnps - если ищем по открытым КВС</li><li>– ddorp или ddorp_all - сли ищем по регистру детей-сирот</li><li>– geriatrics - 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><li>– attbefore3 - только прикрепленные или старше 3 лет</li><li>– att_vol - только прикрепленные и в объеме</li><li>– att_1_4 - только прикрепленные (основное или служебное)</li></ul></li></ul>
---------------------------	---

	<ul style="list-style-type: none"> <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",       "PersonCard_IsDms": "false", </pre>

```
"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,  
"accessType": "edit"

	<pre>         }     ] } </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>

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

Метод возвращает список параметров адреса.

Таблица 6 – Параметры метода **GET api/AddressSearch**

<p><b>Входящие параметры</b></p>	<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.</li> <li>Допустимы форматы: „,,,город,улица,номер дома/здания,,</li> <li>Указанные параметры обязательны. Ведущие/начальные запятые обязательны.</li> <li>– analyze (N) – Признак анализа адреса.</li> </ul> <p>1 – Метод вернёт дополнительную информацию результата анализа поиска адреса.</p>
<p><b>Ответ</b></p>	<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 области/края/республики и т.д.</li> <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> </ul>

- City\_Name (T) – Наименование населённого пункта (города, деревни и т.п.)
- City\_Type (T) – Тип населённого пункта.
- City\_KLADR\_Code (T) – Код КЛАДР населённого пункта.
- City\_Postal\_Index (T) – Почтовый индекс населённого пункта.
- City\_AOGUID (T) – идентификатор адресного объекта по ФИАС (населённого пункта).
- Street\_id (N) – идентификатор улицы.
- Street\_Name (T) – Наименование улицы (переулка, бульвара и т.п.).
- Street\_Type (T) – Тип улицы.
- Street\_KLADR\_Code (T) – Код КЛАДР улицы.
- Street\_Postal\_Index (T) – Почтовый индекс улицы.
- Street\_AOGUID (T) – идентификатор адресного объекта по ФИАС (улицы).
- House (T) – Номер дома (строения, здания и т.п.).
- House\_Corpus (T) – Номер корпуса.
- House\_Type (T) – Тип дома.
- House\_Postal\_Index (T) – Почтовый индекс дома.
- Flat (T) – Номер квартиры.

**Правила:**

- Если найден один адрес, то метод вернёт параметры адреса.
- Если не найдено адреса, то метод вернёт пустое значение.
- Если найдено более одного адреса, то метод вернёт ошибку.
- Если строка адреса не соответствует допустимым форматам, то метод вернёт ошибку.
- В качестве сокращений части адреса допустимы любые

	<p>из справочника dbo.KLSocг с учётом уровня/номера части адреса.</p> <ul style="list-style-type: none"><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>
--	---

## 5.5 Методы сервиса "Запись на вакцинацию"

### 5.5.1 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 7 – 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>
---------------	---

<p><b>Ответ</b></p>	<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>
---------------------	---

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

Таблица 8 – Параметры метода **POST api/TimeTableMedService/TimeTableMedServiceWrite**

<p><b>Входящие параметры</b></p>	<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:</li> <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> </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 Эксплуатация сервиса**

Система предназначена для функционирования 24 часа в сутки 7 дней в неделю. Обеспечивается возможность взаимодействия с пользователями в круглосуточном режиме без перерывов, в том числе при доступе пользователей из других по отношению к серверной части временных зон.

Для программного обеспечения Системы определены следующие режимы функционирования:

- штатный режим (режим, обеспечивающий выполнение функций Системы);
- предаварийный режим (режим, предшествующий переходу в аварийный режим);
- аварийный режим (характеризуется отказом одного или нескольких компонентов программного и/или аппаратного обеспечения. В данном режиме функционируют ресурсы, которые в штатном режиме находятся в режиме горячего резерва)
- сервисный режим (режим для проведения реконфигурирования, обновления и профилактического обслуживания).

Информационный обмен со стороны Системы построен через:

- интеграционную шину Системы с соблюдением правил информационной безопасности;
- Сервисы интеграции.

Подробное описание приведено в документе "Регламент эксплуатации".

## **7 Аварийные ситуации**

### **7.1 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.

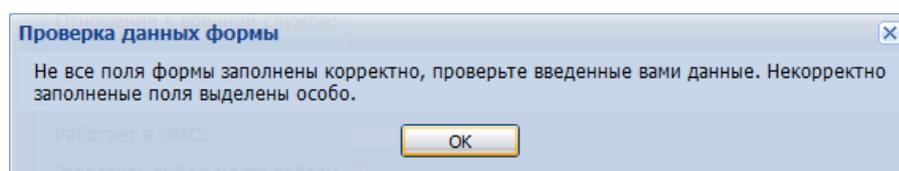
## **7.2 Действия в случае несоблюдения условий выполнения технологического процесса**

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.