|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  | | | |
|  | | | |
| УТВЕРЖДАЮ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. | | УТВЕРЖДАЮ  Генеральный директор ООО «РТ МИС»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Логинов А.В.  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. | |
| **ИНТЕГРАЦИОННЫЕ ПРОФИЛИ**  **Единой цифровой платформы**    Лист утверждения  IX.00.30.01-023 ЛУ | | | |
| Заместитель директора по производству | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | А.А. Марков  «\_\_\_» \_\_\_\_\_\_\_\_\_2020г. |
| Руководитель департамента проектирования и контроля качества | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | А.А. Беляев  «\_\_\_» \_\_\_\_\_\_\_\_\_2020г. |
| Руководитель отдела проектирования | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | О.Н. Зверева  «\_\_\_» \_\_\_\_\_\_\_\_\_2020г. |
| Руководитель отдела документирования | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | А.В. Винокурова  «\_\_\_» \_\_\_\_\_\_\_\_\_2020г. |
| 2020 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  | | | |
|  | | | |
| Утвержден  IX.00.30.01-023 ЛУ | |  | |
| **ИНТЕГРАЦИОННЫЕ ПРОФИЛИ**  **Единой цифровой платформы**  IX.00.30.01-023  Листов 413 | | | |
|  |  | |  |
|  |  | |  |
| 2020 | | | |

**СОДЕРЖАНИЕ**

[ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ 14](#_Toc38975010)

[1 ОБЩАЯ ИНФОРМАЦИЯ 15](#_Toc38975011)

[1.1 Общее описание 15](#_Toc38975014)

[1.1.1 Кэширование запросов 15](#_Toc38975015)

[1.1.2 HTTP ошибки 15](#_Toc38975016)

[1.1.3 Общие параметры для всех методов 15](#_Toc38975017)

[1.1.4 Описание параметров метода 15](#_Toc38975018)

[1.2 Схема взаимодействия 16](#_Toc38975019)

[1.3 Описание общих кодов ошибок 16](#_Toc38975020)

[1.4 Описание последовательности вызова методов при получении запроса на добавление человека 16](#_Toc38975021)

[1.5 Описание последовательности вызова методов при получении документа 21](#_Toc38975022)

[1.6 Сервис извещений об изменениях в расписании, записи на прием, очереди 39](#_Toc38975023)

[1.1.5 Схема взаимодействия 40](#_Toc38975024)

[1.1.6 Используемые методы 40](#_Toc38975025)

[1.1.6.1 Получение данных об изменениях по биркам поликлиники 40](#_Toc38975026)

[1.1.6.2 Получение данных об изменениях в очереди на прием к врачу поликлиники 41](#_Toc38975027)

[2 Системные методы 42](#_Toc38975028)

[2.1 Общие системные методы 42](#_Toc38975031)

[2.1.1 Авторизация в системе 42](#_Toc38975034)

[2.1.2 Выход из системы 42](#_Toc38975035)

[2.1.3 Фильтрация данных и доступ к редактированию данных МО 42](#_Toc38975036)

[2.1.4 Получение списка МО, связанных c пользователем 43](#_Toc38975037)

[2.1.5 Установка МО редактирования/просмотра 43](#_Toc38975038)

[2.1.6 Получение параметров сессии 44](#_Toc38975039)

[2.1.7 Типы справочников, используемых в Системе 44](#_Toc38975040)

[2.1.8 Получение списка справочников 65](#_Toc38975041)

[2.1.9 Получение элементов справочника 66](#_Toc38975042)

[2.1.10 Получение элементов справочника с учетом таблиц стыковки 66](#_Toc38975043)

[2.1.11 Получение элементов справочника СМО (расширенный) 67](#_Toc38975044)

[2.1.12 Получение элементов справочника МО (расширенный) 67](#_Toc38975045)

[2.1.13 Получение элементов справочника территорий КЛАДР (расширенный) 68](#_Toc38975046)

[2.1.14 Получение элементов справочника улиц КЛАДР (расширенный) 69](#_Toc38975047)

[2.1.15 Получение списка районов 69](#_Toc38975048)

[2.1.16 Получение элементов справочника территорий страхования (расширенный) 70](#_Toc38975049)

[2.1.17 Получение элементов справочника услуг 71](#_Toc38975050)

[2.1.18 Получение элементов справочника единиц измерения 71](#_Toc38975051)

[2.1.19 Получение элементов справочника типов льгот 72](#_Toc38975052)

[2.1.20 Получение медикамента МНН 72](#_Toc38975053)

[2.1.21 Получение элементов справочника по столбцу 74](#_Toc38975054)

[2.1.22 Получение элементов справочника по столбцу (расширенный) 74](#_Toc38975055)

[2.1.23 Получение исхода госпитализации по типу подразделения МО 75](#_Toc38975056)

[2.1.24 Получение исхода заболевания по типу подразделения МО 75](#_Toc38975057)

[2.1.25 Получение элементов справочника ЭРСБ с учетом таблицы стыковки 75](#_Toc38975058)

[2.1.26 Получение справочника Статус очереди 76](#_Toc38975059)

[3 Паспорт медицинской организации 77](#_Toc38975060)

[3.1 Описание последовательности вызова методов по Структуре МО 77](#_Toc38975062)

[3.2 Описание последовательности вызова методов по медицинскому персоналу 80](#_Toc38975063)

[3.2.1 Описание последовательности вызова методов по созданию/получению человека 81](#_Toc38975067)

[3.2.2 Описание последовательности вызова методов по созданию/получению сотрудника 83](#_Toc38975068)

[3.2.3 Описание последовательности вызова методов по созданию строки штатного расписания 83](#_Toc38975069)

[3.2.4 Описание последовательности вызова методов по созданию места работы 84](#_Toc38975070)

[3.3 Описание методов работы со Структурой МО 84](#_Toc38975071)

[3.5.1 Получение списка МО региона 84](#_Toc38975075)

[3.5.2 Создание подразделения МО 85](#_Toc38975076)

[3.5.3 Получение списка МО по району 86](#_Toc38975077)

[3.5.4 Получение атрибутов подразделения по идентификатору подразделения 86](#_Toc38975078)

[3.5.5 Получение списка подразделений МО по коду и наименованию 87](#_Toc38975079)

[3.5.6 Изменение данных подразделения 87](#_Toc38975080)

[3.5.7 Создание отделения 88](#_Toc38975081)

[3.5.8 Получение списка отделений МО по коду и наименованию 89](#_Toc38975082)

[3.5.9 Получение списка отделений в МО 90](#_Toc38975083)

[3.5.10 Получение списка отделений в указанном подразделении 91](#_Toc38975084)

[3.5.11 Получение данных отделения по идентификатору 91](#_Toc38975085)

[3.5.12 Изменение данных отделения 92](#_Toc38975086)

[3.5.13 Удаление отделения по идентификатору 93](#_Toc38975087)

[3.5.14 Создание дополнительного профиля 94](#_Toc38975088)

[3.5.15 Получение списка дополнительных профилей отделения 94](#_Toc38975089)

[3.5.16 Изменение дополнительного профиля 94](#_Toc38975090)

[3.5.17 Удаление дополнительного профиля 95](#_Toc38975091)

[3.5.18 Создание палаты в отделении 95](#_Toc38975092)

[3.5.19 Получение списка палат отделения по наименованию 96](#_Toc38975093)

[3.5.20 Изменение палаты 96](#_Toc38975094)

[3.5.21 Удаление палаты по идентификатору 97](#_Toc38975095)

[3.5.22 Получение списка палат в отделении 97](#_Toc38975096)

[3.5.23 Получение атрибутов палаты по идентификатору 97](#_Toc38975097)

[3.5.24 Добавление объектов комфортности в палате 98](#_Toc38975098)

[3.5.25 Получение объектов комфортности палаты по наименованию 98](#_Toc38975099)

[3.5.26 Изменение объектов комфортности в палате 98](#_Toc38975100)

[3.5.27 Получение списка объектов комфортности по палате 98](#_Toc38975101)

[3.5.28 Удаление объектов комфортности по идентификатору 99](#_Toc38975102)

[3.5.29 Создание участка 99](#_Toc38975103)

[3.5.30 Получение списка участков по номеру 99](#_Toc38975104)

[3.5.31 Редактирование участка 100](#_Toc38975105)

[3.5.32 Получение общих данных участка по идентификатору 100](#_Toc38975106)

[3.5.33 Получение списка участков по МО 100](#_Toc38975107)

[3.5.34 Создание периода работы врача на участке 101](#_Toc38975108)

[3.5.35 Изменение периода работы врача на участке 101](#_Toc38975109)

[3.5.36 Получение информации о периоде работы врача на участке по датам начала и окончания 101](#_Toc38975110)

[3.5.37 Получение информации о периоде работы по идентификатору 102](#_Toc38975111)

[3.5.38 Получение периодов работы врачей на участке 102](#_Toc38975112)

[3.5.39 Получение списка периодов работы врачей по рабочему месту врача 102](#_Toc38975113)

[3.5.40 Создание группы отделений МО 103](#_Toc38975114)

[3.5.41 Изменение группы отделений МО 104](#_Toc38975115)

[3.5.42 Получение данных о группе отделений МО по идентификатору 105](#_Toc38975116)

[3.5.43 Получение списка групп отделений МО по идентификатору подразделения 105](#_Toc38975117)

[3.5.44 Создание коек в Коечном фонде отделения 106](#_Toc38975118)

[3.5.45 Изменение коек в коечном фонде отделения 107](#_Toc38975119)

[3.5.46 Удаление в отделении МО коек по профилю 107](#_Toc38975120)

[3.5.47 Получение списка коек коечного фонда отделения МО 108](#_Toc38975121)

[3.5.48 Получение списка служб по МО 108](#_Toc38975122)

[3.4 Описание методов работы с сотрудниками МО 109](#_Toc38975123)

[3.6.1 Создание сотрудника 109](#_Toc38975125)

[3.6.2 Получение данных сотрудника по идентификатору человека 109](#_Toc38975126)

[3.6.3 Получение данных сотрудника по идентификатору сотрудника 109](#_Toc38975127)

[3.6.4 Получение мест работы медицинского работника (на данный момент) 110](#_Toc38975128)

[3.6.5 Получение данных о местах работы сотрудника 110](#_Toc38975129)

[3.6.6 Создание данных о среднем или профессиональном образовании сотрудника 111](#_Toc38975130)

[3.6.7 Изменение данных о среднем или профессиональном образовании сотрудника 112](#_Toc38975131)

[3.6.8 Удаление данных о среднем или профессиональном образовании сотрудника 112](#_Toc38975132)

[3.6.9 Получение данных о среднем или профессиональном образовании сотрудника 113](#_Toc38975133)

[3.6.10 Создание данных о послевузовском образовании сотрудника 113](#_Toc38975134)

[3.6.11 Изменение данных о послевузовском образовании сотрудника 114](#_Toc38975135)

[3.6.12 Удаление данных о послевузовском образовании сотрудника 115](#_Toc38975136)

[3.6.13 Получение данных о послевузовском образовании сотрудника 115](#_Toc38975137)

[3.6.14 Создание данных о курсах переподготовки сотрудника 116](#_Toc38975138)

[3.6.15 Изменение данных о курсах переподготовки сотрудника 117](#_Toc38975139)

[3.6.16 Удаление данных о курсах переподготовки сотрудника 117](#_Toc38975140)

[3.6.17 Получение данных о курсах переподготовки сотрудника 118](#_Toc38975141)

[3.6.18 Создание данных о курсах повышения квалификации 118](#_Toc38975142)

[3.6.19 Изменение данных о курсах повышения квалификации 119](#_Toc38975143)

[3.6.20 Удаление данных о курсах повышения квалификации 119](#_Toc38975144)

[3.6.21 Получение данных о курсах повышения квалификации 120](#_Toc38975145)

[3.6.22 Создание данных о сертификатах сотрудника 120](#_Toc38975146)

[3.6.23 Изменение данных о сертификатах сотрудника 121](#_Toc38975147)

[3.6.24 Удаление данных о сертификатах сотрудника 121](#_Toc38975148)

[3.6.25 Получение данных о сертификатах сотрудника 121](#_Toc38975149)

[3.6.26 Создание данных о квалификационных категориях сотрудника 122](#_Toc38975150)

[3.6.27 Изменение данных о квалификационных категориях (сертификатах) сотрудника 123](#_Toc38975151)

[3.6.28 Удаление данных о квалификационных категориях сотрудника 123](#_Toc38975152)

[3.6.29 Получение данных о квалификационных категориях сотрудника 123](#_Toc38975153)

[3.6.30 Создание данных о наградах сотрудника 124](#_Toc38975154)

[3.6.31 Изменение данных о наградах сотрудника 124](#_Toc38975155)

[3.6.32 Удаление данных о наградах сотрудника 125](#_Toc38975156)

[3.6.33 Получение данных о наградах сотрудника 125](#_Toc38975157)

[3.6.34 Создание данных об аккредитации сотрудника 125](#_Toc38975158)

[3.6.35 Изменение данных об аккредитации сотрудника 126](#_Toc38975159)

[3.6.36 Удаление данных об аккредитации сотрудника 127](#_Toc38975160)

[3.6.37 Получение данных об аккредитации сотрудника 127](#_Toc38975161)

[3.5 Описание методов работы с должностью 128](#_Toc38975162)

[3.7.1 Создание должности 128](#_Toc38975164)

[3.7.2 Изменение должности 128](#_Toc38975165)

[3.7.3 Получение должности по коду 128](#_Toc38975166)

[3.7.4 Получение должности по идентификатору 129](#_Toc38975167)

[3.6 Описание методов работы со штатным расписанием 129](#_Toc38975168)

[3.8.1 Создание строки штатного расписания 129](#_Toc38975170)

[3.8.2 Изменение строки штатного расписания 130](#_Toc38975171)

[3.8.3 Получение строки штатного расписания по идентификатору 130](#_Toc38975172)

[3.8.4 Получение строки штатного расписания по месту работы 131](#_Toc38975173)

[3.8.5 Получение строки штатного расписания по отделению МО 131](#_Toc38975174)

[3.8.6 Создание выплаты для строки штатного расписания 131](#_Toc38975175)

[3.8.7 Изменение выплаты 132](#_Toc38975176)

[3.8.8 Получение выплат по строке штатного расписания 132](#_Toc38975177)

[3.7 Описание методов работы с местом работы 132](#_Toc38975178)

[3.9.1 Создание места работы 132](#_Toc38975180)

[3.9.2 Изменение места работы 133](#_Toc38975181)

[3.9.3 Получение места работы по идентификатору 134](#_Toc38975182)

[3.9.4 Получение списка мест работы по МО и профилю 136](#_Toc38975183)

[3.9.5 Создание невыплаты 136](#_Toc38975184)

[3.9.6 Изменение невыплаты 136](#_Toc38975185)

[3.9.7 Получение невыплаты по идентификатору 137](#_Toc38975186)

[3.9.8 Получение навыплат по месту работы 137](#_Toc38975187)

[3.9.9 Создание выплаты для места работы 137](#_Toc38975188)

[3.9.10 Получение выплат по месту работы 137](#_Toc38975189)

[3.8 Описание методов работы с паспортом МО 137](#_Toc38975190)

[3.10.1 Изменение паспорта МО 137](#_Toc38975192)

[3.10.2 Добавление периода по ОМС 139](#_Toc38975193)

[3.10.3 Изменение периода по ОМС 139](#_Toc38975194)

[3.10.4 Получение списка периодов по ОМС в МО 139](#_Toc38975195)

[3.10.5 Получение атрибутов периода по ОМС по идентификатору 139](#_Toc38975196)

[3.10.6 Удаление периода по ОМС 140](#_Toc38975197)

[3.10.7 Добавление периода по ЛЛО для МО 140](#_Toc38975198)

[3.10.8 Изменение периода по ЛЛО для МО 140](#_Toc38975199)

[3.10.9 Получение списка периодов по ЛЛО в МО 140](#_Toc38975200)

[3.10.10 Получение атрибутов периода по ЛЛО по идентификатору 141](#_Toc38975201)

[3.10.11 Удаление периода по ЛЛО 141](#_Toc38975202)

[3.10.12 Добавление информационной системы 141](#_Toc38975203)

[3.10.13 Изменение информационной системы 141](#_Toc38975204)

[3.10.14 Получение списка информационных систем по МО 142](#_Toc38975205)

[3.10.15 Получение атрибутов информационной системы по идентификатору 142](#_Toc38975206)

[3.10.16 Удаление информационной системы 143](#_Toc38975207)

[3.10.17 Добавление специализации организации 143](#_Toc38975208)

[3.10.18 Изменение специализации организации 143](#_Toc38975209)

[3.10.19 Получение списка специализаций по МО 143](#_Toc38975210)

[3.10.20 Получение атрибутов специализации по идентификатору 144](#_Toc38975211)

[3.10.21 Удаление специализации организации 144](#_Toc38975212)

[3.10.22 Добавление лицензии МО 144](#_Toc38975213)

[3.10.23 Изменение лицензии МО 145](#_Toc38975214)

[3.10.24 Получение лицензии по МО, номеру лицензии, выдавшей организации и дате выдачи 145](#_Toc38975215)

[3.10.25 Получение списка лицензий по МО 145](#_Toc38975216)

[3.10.26 Получение атрибутов лицензии МО по идентификатору 146](#_Toc38975217)

[3.10.27 Удаление лицензии МО 146](#_Toc38975218)

[3.10.28 Добавление вида лицензии по профилю 146](#_Toc38975219)

[3.10.29 Изменение вида лицензии по профилю 146](#_Toc38975220)

[3.10.30 Получение списка видов профилей по лицензии 147](#_Toc38975221)

[3.10.31 Удаление вида лицензии по профилю 147](#_Toc38975222)

[3.10.32 Добавление операции, проведенной с лицензией 147](#_Toc38975223)

[3.10.33 Изменение операции, проведенной с лицензией 147](#_Toc38975224)

[3.10.34 Получение списка операций, проведенных с лицензией 148](#_Toc38975225)

[3.10.35 Удаление операции, проведенной с лицензией 148](#_Toc38975226)

[3.10.36 Добавление медтехнологии 148](#_Toc38975227)

[3.10.37 Изменение медтехнологии 148](#_Toc38975228)

[3.10.38 Получение списка медтехнологий по МО 149](#_Toc38975229)

[3.10.39 Получение списка медтехнологий по идентификатору здания 149](#_Toc38975230)

[3.10.40 Получение атрибутов медтехнологии по идентификатору 149](#_Toc38975231)

[3.10.41 Удаление медтехнологии 150](#_Toc38975232)

[3.10.42 Добавление медицинской услуги в паспорт МО 150](#_Toc38975233)

[3.10.43 Изменение медицинской услуги в паспорте МО 150](#_Toc38975234)

[3.10.44 Получение списка медицинских услуг по МО 150](#_Toc38975235)

[3.10.45 Получение атрибутов медицинской услуги по идентификатору 150](#_Toc38975236)

[3.10.46 Удаление медицинской услуги из паспорта МО 151](#_Toc38975237)

[3.10.47 Добавление направления оказания медицинской помощи 151](#_Toc38975238)

[3.10.48 Изменение направления оказания медицинской помощи 151](#_Toc38975239)

[3.10.49 Получение списка направлений оказания медицинской помощи 152](#_Toc38975240)

[3.10.50 Получение атрибутов направления оказания медицинской помощи по идентификатору 152](#_Toc38975241)

[3.10.51 Удаление направления оказания медицинской помощи 152](#_Toc38975242)

[3.10.52 Добавление объекта/места использования природного лечебного фактора 152](#_Toc38975243)

[3.10.53 Изменение объекта/места использования природного лечебного фактора 153](#_Toc38975244)

[3.10.54 Получение списка объектов/мест использования природного лечебного фактора по МО 153](#_Toc38975245)

[3.10.55 Получение атрибутов объекта/места использования природного лечебного фактора по идентификатору 153](#_Toc38975246)

[3.10.56 Удаление объекта/места использования природного лечебного фактора 153](#_Toc38975247)

[3.10.57 Добавление периода функционирования МО 154](#_Toc38975248)

[3.10.58 Изменение периода функционирования МО 154](#_Toc38975249)

[3.10.59 Получение списка периодов функционирования по МО 154](#_Toc38975250)

[3.10.60 Получение атрибутов периода функционирования по идентификатору 155](#_Toc38975251)

[3.10.61 Удаление периода функционирования МО 155](#_Toc38975252)

[3.10.62 Добавление питания 155](#_Toc38975253)

[3.10.63 Изменение питания 156](#_Toc38975254)

[3.10.64 Получение списка питаний по МО 156](#_Toc38975255)

[3.10.65 Получение атрибутов питания по идентификатору 156](#_Toc38975256)

[3.10.66 Удаление питания 157](#_Toc38975257)

[3.10.67 Добавление природного лечебного фактора 157](#_Toc38975258)

[3.10.68 Изменение природного лечебного фактора 157](#_Toc38975259)

[3.10.69 Получение списка природных лечебных факторов 158](#_Toc38975260)

[3.10.70 Получение атрибутов природного лечебного фактора по идентификатору 158](#_Toc38975261)

[3.10.71 Удаление природного лечебного фактора 158](#_Toc38975262)

[3.10.72 Добавление расчетного счета 158](#_Toc38975263)

[3.10.73 Изменение расчетного счета 159](#_Toc38975264)

[3.10.74 Получение списка расчетных счетов по МО 159](#_Toc38975265)

[3.10.75 Получение атрибутов расчетного счета по идентификатору 160](#_Toc38975266)

[3.10.76 Удаление расчетного счета 160](#_Toc38975267)

[3.10.77 Добавление КБК для расчетного счета 160](#_Toc38975268)

[3.10.78 Изменение КБК для расчетного счета 160](#_Toc38975269)

[3.10.79 Получение списка КБК по расчетному счету 161](#_Toc38975270)

[3.10.80 Удаление КБК для расчетного счета 161](#_Toc38975271)

[3.10.81 Добавление территории обслуживания 161](#_Toc38975272)

[3.10.82 Изменение территории обслуживания 162](#_Toc38975273)

[3.10.83 Получение списка территорий обслуживания 162](#_Toc38975274)

[3.10.84 Получение атрибутов территории обслуживания по идентификатору 162](#_Toc38975275)

[3.10.85 Удаление территории обслуживания 163](#_Toc38975276)

[3.10.86 Добавление руководства организации 163](#_Toc38975277)

[3.10.87 Изменение руководства организации 163](#_Toc38975278)

[3.10.88 Получение списка руководства по МО 164](#_Toc38975279)

[3.10.89 Получение руководства организации по идентификатору 164](#_Toc38975280)

[3.10.90 Удаление руководства организации 164](#_Toc38975281)

[3.10.91 Добавление заезда 164](#_Toc38975282)

[3.10.92 Изменение заезда 165](#_Toc38975283)

[3.10.93 Получение списка заездов по МО 165](#_Toc38975284)

[3.10.94 Получение атрибутов заезда по идентификатору 165](#_Toc38975285)

[3.10.95 Удаление заезда 165](#_Toc38975286)

[3.10.96 Добавление округа горно-санитарной охраны 166](#_Toc38975287)

[3.10.97 Изменение округа горно-санитарной охраны 166](#_Toc38975288)

[3.10.98 Получение списка округов горно-санитарной охраны 166](#_Toc38975289)

[3.10.99 Получение атрибутов округа горно-санитарной охраны по идентификатору 167](#_Toc38975290)

[3.10.100 Удаление округа горно-санитарной охраны 167](#_Toc38975291)

[3.10.101 Добавление статуса курорта 167](#_Toc38975292)

[3.10.102 Изменение статуса курорта 167](#_Toc38975293)

[3.10.103 Получение списка статусов курорта по МО 168](#_Toc38975294)

[3.10.104 Получение атрубутов статуса курорта по идентификатору 168](#_Toc38975295)

[3.10.105 Удаление статуса курорта 168](#_Toc38975296)

[3.10.106 Добавление типа курорта 169](#_Toc38975297)

[3.10.107 Изменение типа курорта 169](#_Toc38975298)

[3.10.108 Получение списка типов курорта по МО 169](#_Toc38975299)

[3.10.109 Получение типа курорта по идентификатору 169](#_Toc38975300)

[3.10.110 Удаление типа курорта 170](#_Toc38975301)

[3.10.111 Добавление объекта инфраструктуры 170](#_Toc38975302)

[3.10.112 Изменение объекта инфраструктуры 170](#_Toc38975303)

[3.10.113 Получение списка объектов инфраструктуры по МО 171](#_Toc38975304)

[3.10.114 Получение объекта инфраструктуры по идентификатору 171](#_Toc38975305)

[3.10.115 Удаление объекта инфраструктуры 171](#_Toc38975306)

[3.10.116 Добавление площадки, занимаемой организацией 171](#_Toc38975307)

[3.10.117 Изменение площадки, занимаемой организацией 172](#_Toc38975308)

[3.10.118 Получение списка площадок, занимаемых организацией 172](#_Toc38975309)

[3.10.119 Получение идентификатора площадки по наименованию площадки и идентификатору участка 173](#_Toc38975310)

[3.10.120 Получение атрибутов площадки, занимаемой организацией, по идентификатору 173](#_Toc38975311)

[3.10.121 Удаление площадки, занимаемой организацией 173](#_Toc38975312)

[3.10.122 Добавление связи с транспортными узлами 173](#_Toc38975313)

[3.10.123 Изменение связи с транспортными узлами 174](#_Toc38975314)

[3.10.124 Получение списка связей площадки с транспортными узлами 175](#_Toc38975315)

[3.10.125 Получение атрибутов связи с транспортными узлами по идентификатору. 175](#_Toc38975316)

[3.10.126 Удаление связи с транспортными узлами 175](#_Toc38975317)

[3.10.127 Добавление здания МО 176](#_Toc38975318)

[3.10.128 Изменение атрибутов здания МО 178](#_Toc38975319)

[3.10.129 Получение списка зданий МО 181](#_Toc38975320)

[3.10.130 Получение здания МО по номеру учета 181](#_Toc38975321)

[3.10.131 Получение атрибутов здания МО по идентификатору 182](#_Toc38975322)

[3.10.132 Удаление здания МО 184](#_Toc38975323)

[3.10.133 Добавление Домового хозяйства 184](#_Toc38975324)

[3.10.134 Изменение Домового хозяйства 184](#_Toc38975325)

[3.10.135 Удаление Домового хозяйства 185](#_Toc38975326)

[3.10.136 Получение списка Домовых хозяйств МО 185](#_Toc38975327)

[3.9 Описание методов работы с медицинскими изделиями 186](#_Toc38975328)

[3.11.1 Получение Медицинского изделия по идентификатору 186](#_Toc38975330)

[3.11.2 Создание Медицинского изделия 188](#_Toc38975331)

[3.11.3 Редактирование Медицинского изделия 190](#_Toc38975332)

[3.11.4 Получение списка Медицинских изделий одного Класса 191](#_Toc38975333)

[3.11.5 Подсчет количества Медицинских изделий одного Класса 192](#_Toc38975334)

[3.11.6 Получение «Классификатор Медицинского изделия» 192](#_Toc38975335)

[3.11.7 Создание «Классификатор Медицинского изделия» 193](#_Toc38975336)

[3.11.8 Редактирование «Классификатор Медицинского изделия» 193](#_Toc38975337)

[3.11.9 Получение записи «Расходные материалы» по идентификатору 194](#_Toc38975338)

[3.11.10 Получение записи «Расходные материалы» по наименованию и мед изделию 194](#_Toc38975339)

[3.11.11 Создание записи «Расходные материалы» 194](#_Toc38975340)

[3.11.12 Редактирование записи «Расходные материалы» 195](#_Toc38975341)

[3.11.13 Получение записи «Свидетельство о проверке» 195](#_Toc38975342)

[3.11.14 Создание записи «Свидетельство о проверке» 195](#_Toc38975343)

[3.11.15 Редактирование записи «Свидетельство о проверке» 196](#_Toc38975344)

[3.11.16 Получение записи «Начисление износа» 196](#_Toc38975345)

[3.11.17 Создание записи «Начисление износа» 196](#_Toc38975346)

[3.11.18 Редактирование записи «Начисление износа» 196](#_Toc38975347)

[3.11.19 Получение записи «Простой МИ» 197](#_Toc38975348)

[3.11.20 Получение списка записей «Простой МИ» для МИ 197](#_Toc38975349)

[3.11.21 Создание записи «Простой МИ» 197](#_Toc38975350)

[3.11.22 Редактирование записи «Простой МИ» 198](#_Toc38975351)

[3.11.23 Получение записи «Эксплуатационные данные» 198](#_Toc38975352)

[3.11.24 Создание записи «Эксплуатационные данные» 198](#_Toc38975353)

[3.11.25 Редактирование записи «Эксплуатационные данные» 199](#_Toc38975354)

[3.11.26 Удаление свойств Медицинского изделия 199](#_Toc38975355)

[3.10 Описание методов работы с данными организации 199](#_Toc38975356)

[3.12.1 Поиск организации по параметрам 199](#_Toc38975358)

[3.12.2 Создание записи «Организация» 200](#_Toc38975359)

[3.12.3 Редактирование записи «Организация» 200](#_Toc38975360)

[3.12.4 Получение записи «Адрес» для организации 200](#_Toc38975361)

[3.12.5 Создание записи «Адрес» 201](#_Toc38975362)

[3.12.6 Редактирование записи «Адрес» 201](#_Toc38975363)

[3.12.7 Получение кода для новой организации 201](#_Toc38975364)

[4 Электронная медицинская карта 202](#_Toc38975365)

[4.1 Описание методов работы с ТАП 202](#_Toc38975367)

[4.1.1 Создание ТАП с первым посещением для поликлинического случая 202](#_Toc38975370)

[4.1.2 Добавление посещения в ТАП Поликлинический случай 203](#_Toc38975371)

[4.1.3 Редктирование ТАП Поликлинический случай 204](#_Toc38975372)

[4.1.4 Редактирование Посещения поликлинический случай 204](#_Toc38975373)

[4.1.5 Получить ТАП 205](#_Toc38975374)

[4.1.6 Получить посещение 206](#_Toc38975375)

[4.1.7 Получить список посещений для ТАП 207](#_Toc38975376)

[4.1.8 Получить данные ТАП (стоматология) по идентификатору 207](#_Toc38975377)

[4.1.9 Получить список посещений для ТАП (стоматология) 208](#_Toc38975378)

[4.1.10 Получить список заболеваний для ТАП (стоматология) 208](#_Toc38975379)

[4.1.11 Получить посещение по идентификатору(стоматология) 208](#_Toc38975380)

[4.1.12 Получить заболевание по идентификатору (стоматология) 209](#_Toc38975381)

[4.1.13 Создать ТАП стоматологический 209](#_Toc38975382)

[4.1.14 Установка заболевания (ТАП стоматологический) 209](#_Toc38975383)

[4.1.15 Создать посещение (ТАП стоматологический) 210](#_Toc38975384)

[4.1.16 Изменить посещение (ТАП стоматологический) 210](#_Toc38975385)

[4.1.17 Изменить заболевание (ТАП стоматологический) 211](#_Toc38975386)

[4.1.18 Получить информацию об оказанной услуги по идентификатору (стоматология) 211](#_Toc38975387)

[4.1.19 Получить список оказанных услуг в рамках заболевания (стоматология) 211](#_Toc38975388)

[4.1.20 Получить список оказанных услуг в рамках посещения (стоматология) 212](#_Toc38975389)

[4.1.21 Создать услугу в Стоматологии 212](#_Toc38975390)

[4.1.22 Изменить услугу в Стоматологии 212](#_Toc38975391)

[4.1.23 Добавить сопутствующий диагноз для заболевания 212](#_Toc38975392)

[4.1.24 Получить список сопутствующих диагнозов для заболевания 213](#_Toc38975393)

[4.1.25 Получить сопутствующий диагноз для заболевания по идентификатору 213](#_Toc38975394)

[4.1.26 Изменить сопутствующий диагноз для заболевания 213](#_Toc38975395)

[4.1.27 Получение информации по ТАП (закрытые/открытые) 213](#_Toc38975396)

[4.1.28 Получение информации по ТАП (стоматология) (закрытые/открытые) 214](#_Toc38975397)

[4.1.29 Получение информации по случаю амбулаторно-поликлинического лечения 214](#_Toc38975398)

[4.2 Описание методов работы с услугами 217](#_Toc38975399)

[4.2.1 Получение списка выполненных услуг 217](#_Toc38975401)

[4.2.2 Получение информации по оперативной услуге 218](#_Toc38975402)

[4.2.3 Получение информации по общей услуге 219](#_Toc38975403)

[4.2.4 Получение информации по параклинической услуге 220](#_Toc38975404)

[4.2.5 Получение информации об услуге по типу услуги 221](#_Toc38975405)

[4.2.6 Получение информации об услуге по идентификатору направления 222](#_Toc38975406)

[4.2.7 Получение информации об услуге по идентификатору направления 222](#_Toc38975407)

[4.2.8 Создание оперативной услуги 222](#_Toc38975408)

[4.2.9 Создание общей услуги 224](#_Toc38975409)

[4.2.10 Создание параклинической услуги 225](#_Toc38975410)

[4.2.11 Редактирование оперативной услуги 225](#_Toc38975411)

[4.2.12 Редактирование общей услуги 227](#_Toc38975412)

[4.2.13 Редактирование параклинической услуги 227](#_Toc38975413)

[4.2.14 Получение списка анестезии по услуге 228](#_Toc38975414)

[4.2.15 Получение информации по анестезии 228](#_Toc38975415)

[4.2.16 Создание анестезии на оперативной услуге 229](#_Toc38975416)

[4.2.17 Редактирование анестезии на оперативной услуге 229](#_Toc38975417)

[4.2.18 Получение списка осложнений на услуге 229](#_Toc38975418)

[4.2.19 Получение информации по осложнению услуги 230](#_Toc38975419)

[4.2.20 Создание осложнения на услуге 230](#_Toc38975420)

[4.2.21 Редактирование осложнения на услуге 231](#_Toc38975421)

[4.3 Описание методов работы с документами нетрудоспособности 231](#_Toc38975422)

[4.3.1 Получение списка документов нетрудоспособности по случаю лечения 231](#_Toc38975424)

[4.3.2 Получение информации по ЛВН 231](#_Toc38975425)

[4.3.3 Получение информации по Справке учащегося 232](#_Toc38975426)

[4.3.4 Создание ЛВН 233](#_Toc38975427)

[4.3.5 Создание справки учащегося 234](#_Toc38975428)

[4.3.6 Редактирование информации по ЛВН 235](#_Toc38975429)

[4.3.7 Редактирование информации по справке учащегося 236](#_Toc38975430)

[4.3.8 Получение списка периодов освобождения от работы 236](#_Toc38975431)

[4.3.9 Получение информации по периоду освобождения от работы 237](#_Toc38975432)

[4.3.10 Создание периода освобождения от работы 237](#_Toc38975433)

[4.3.11 Редактирование информации по периоду освобождения от работы 238](#_Toc38975434)

[4.3.12 Получение списка связей случая лечения с ЛВН 238](#_Toc38975435)

[4.3.13 Создание связи случая лечения с ЛВН 239](#_Toc38975436)

[4.3.14 Удаление связи случая лечения с ЛВН 239](#_Toc38975437)

[4.4 Описание методов работы со свидетельствами 239](#_Toc38975438)

[4.4.1 Получение списка свидетельств о рождении по человеку 239](#_Toc38975440)

[4.4.2 Получение информации по Свидетельству о рождении 240](#_Toc38975441)

[4.4.3 Создание Свидетельства о рождении 241](#_Toc38975442)

[4.4.4 Редактирование информации по Свидетельству о рождении 243](#_Toc38975443)

[4.4.5 Получение списка свидетельств о смерти по человеку 245](#_Toc38975444)

[4.4.6 Получение информации по Свидетельству о смерти 245](#_Toc38975445)

[4.4.7 Создание Свидетельства о смерти 247](#_Toc38975446)

[4.4.8 Редактирование информации по Свидетельству о смерти 251](#_Toc38975447)

[4.4.9 Получение списка свидетельств о перинатальной смерти по человеку 255](#_Toc38975448)

[4.4.10 Получение информации по Свидетельству о перинатальной смерти 255](#_Toc38975449)

[4.4.11 Создание Свидетельства о перинатальной смерти 257](#_Toc38975450)

[4.4.12 Редактирование информации по Свидетельству о перинатальной смерти 259](#_Toc38975451)

[4.5 Описание методов работы с прочими документами (шаблоны, эпикризы) 261](#_Toc38975452)

[4.5.1 Получение списка документов случая 261](#_Toc38975454)

[4.5.2 Получение содержимого документа 262](#_Toc38975455)

[4.5.3 Создание документа 262](#_Toc38975456)

[4.5.4 Редактирование документа 262](#_Toc38975457)

[4.6 Описание методов работы с назначениями медикаментов 263](#_Toc38975458)

[4.6.1 Получение списка назначений медикаментов в рамках случая 263](#_Toc38975460)

[4.6.2 Получение информации по назначению медикамента 263](#_Toc38975461)

[4.6.3 Создание назначения медикамента 264](#_Toc38975462)

[4.6.4 Редактирование назначения медикамента 264](#_Toc38975463)

[4.6.5 Описание методов работы с назначениями в КВС/ТАП 265](#_Toc38975464)

[4.6.5.1 Назначение режима 265](#_Toc38975465)

[4.6.5.1.1. Получить назначение по идентификатору назначения 265](#_Toc38975466)

[4.6.5.1.2. Создать назначение для ТАП/КВС 265](#_Toc38975467)

[4.7 Описание методов работы с КВС 266](#_Toc38975468)

[4.7.1 Получить КВС 266](#_Toc38975470)

[4.7.2 Получение списка движений для КВС 267](#_Toc38975471)

[4.7.3 Получение списка диагнозов (основного, сопутствующих или осложнений) по движению 268](#_Toc38975472)

[4.7.4 Получить движение 268](#_Toc38975473)

[4.7.5 Получить установленный диагноз в стационаре (сопутсвтующий, осложнение) по идентификатору диагноза 269](#_Toc38975474)

[4.7.6 Создать КВС (c приемным) 270](#_Toc38975475)

[4.7.7 Создать движение в отделение 271](#_Toc38975476)

[4.7.8 Установка диагноза пациенту в стационаре: Добавление (Сопутствующие диагнозы) или осложнения 272](#_Toc38975477)

[4.7.9 Редактировать КВС 273](#_Toc38975478)

[4.7.10 Редактировать движение в отделение 274](#_Toc38975479)

[4.7.11 Редактирование диагноза пациенту в стационаре: сопутствующие диагнозы или осложнения 275](#_Toc38975480)

[4.8 Описание методов работы с выпиской льготных рецептов 275](#_Toc38975481)

[4.8.1 Получение списка выписанных льготных рецептов 275](#_Toc38975483)

[4.8.2 Получение информации по выписанному льготному рецепту 276](#_Toc38975484)

[4.8.3 Создание случая выписки льготного рецепта 277](#_Toc38975485)

[4.8.4 Редактирование случая выписки льготного рецепта 278](#_Toc38975486)

[4.9 Описание методов работы с Направлением 278](#_Toc38975487)

[4.9.1 Получение списка направлений 278](#_Toc38975489)

[4.9.2 Получение информации по направлению 279](#_Toc38975490)

[4.9.3 Создание направления 281](#_Toc38975491)

[4.9.4 Отмена направления 283](#_Toc38975492)

[4.9.5 Получение атрибутов стационарной бирки по идентификатору 283](#_Toc38975493)

[4.9.6 Получение атрибутов бирки на службу по идентификатору 283](#_Toc38975494)

[4.9.7 Получение атрибутов бирки на ресурс по идентификатору 284](#_Toc38975495)

[4.10 Описание методов работы с назначениями лекарственных средств 284](#_Toc38975496)

[4.10.1 Получение назначений лекарственных средств 284](#_Toc38975498)

[4.10.2 Создание назначения лекарственного средства 286](#_Toc38975499)

[4.10.3 Редактирование назначения лекарственного средства 287](#_Toc38975500)

[4.11 Описание методов работы с результатами диагностических исследований 289](#_Toc38975501)

[4.11.1 Добавление исследования, прикрепленного к услуге 289](#_Toc38975503)

[4.11.2 Изменение исследования, прикрепленного к услуге 289](#_Toc38975504)

[4.11.3 Удаление исследования, прикрепленного к услуге 289](#_Toc38975505)

[4.11.4 Получение результатов исследования по параклинической услуге 290](#_Toc38975506)

[4.11.5 Описание обработки данных о врачах, выполняющих лабораторные исследования 290](#_Toc38975507)

[4.11.6 Добавление факта взятия пробы 290](#_Toc38975508)

[4.11.7 Изменение результатов взятия пробы 291](#_Toc38975509)

[4.11.8 Добавление результатов лабораторного исследования 292](#_Toc38975510)

[4.11.9 Изменение результатов лабораторного исследования 293](#_Toc38975511)

[4.11.10 Добавление прикрепленных файлов лабораторного исследования 293](#_Toc38975512)

[4.11.11 Получение прикрепленных файлов лабораторного исследования 294](#_Toc38975513)

[4.11.12 Удаление файлов, прикрепленных к исследованию 294](#_Toc38975514)

[5 Запись на прием к врачу 295](#_Toc38975515)

[5.1 Описание последовательности вызова методов по созданию расписания и записи к врачу 295](#_Toc38975521)

[5.2 Описание REST методов работы с расписанием 301](#_Toc38975522)

[5.2.1 Получение списка специальностей в МО 301](#_Toc38975523)

[5.2.2 Получение списка мест работы по специальности и МО 301](#_Toc38975524)

[5.2.3 Получение свободных дат приема 302](#_Toc38975525)

[5.2.4 Получение свободного времени приема 303](#_Toc38975526)

[5.2.5 Запись пациента на прием 303](#_Toc38975527)

[5.2.6 Получение записей на прием по МО 303](#_Toc38975528)

[5.2.7 Получение списка записанных в МО 304](#_Toc38975529)

[5.2.8 Добавление расписания врача 304](#_Toc38975530)

[5.2.9 Изменение расписания врача 305](#_Toc38975531)

[5.2.10 Получение атрибутов бирки по идентификатору 305](#_Toc38975532)

[5.2.11 Получение статуса записи на прием 305](#_Toc38975533)

[5.2.12 Изменение статуса записи на прием 305](#_Toc38975534)

[5.2.13 Добавление записи в лист ожидания 306](#_Toc38975535)

[5.2.14 Изменение статуса записи в листе ожидания 307](#_Toc38975536)

[5.2.15 Изменение листа ожидания 308](#_Toc38975537)

[5.2.16 Добавление примечания к расписанию 309](#_Toc38975538)

[5.2.17 Изменение примечания к расписанию 309](#_Toc38975539)

[5.2.18 Получение листа ожидания 309](#_Toc38975540)

[5.3 Описание методов работы с вызовом врача на дом 310](#_Toc38975541)

[5.3.1 Добавление вызова врача на дом 310](#_Toc38975546)

[5.3.2 Изменение статуса вызова врача на дом 311](#_Toc38975547)

[5.3.3 Получение атрибутов вызова врача на дом по идентификатору 311](#_Toc38975548)

[5.3.4 Отмена вызова 312](#_Toc38975549)

[5.3.5 Получение симптомов вызова 312](#_Toc38975550)

[5.4 Описание методов работы с адресом 313](#_Toc38975551)

[5.4.1 Получение адреса 313](#_Toc38975553)

[5.4.2 Создание адреса 313](#_Toc38975554)

[5.4.3 Редактирование адреса 315](#_Toc38975555)

[6 Описание методов работы с человеком 316](#_Toc38975556)

[6.1 Получение списка физических лиц, по ключевым параметрам 316](#_Toc38975559)

[6.2 Получение информации по человеку 316](#_Toc38975560)

[6.3 Создание человека 317](#_Toc38975561)

[6.4 Редактирование человека 318](#_Toc38975562)

[6.5 Получение полисных данных пациента 319](#_Toc38975563)

[6.6 Создание полиса 320](#_Toc38975564)

[6.7 Изменение полисных данных человека 321](#_Toc38975565)

[6.8 Получение документа удостоверяющего личность 322](#_Toc38975566)

[6.9 Создание документа 322](#_Toc38975567)

[6.10 Изменение документа 323](#_Toc38975568)

[6.11 Получение ИИН и РПН идентификатор человека 325](#_Toc38975569)

[6.12 Получение записей на прием по МО 325](#_Toc38975570)

[6.13 Получение записей на прием к врачу 325](#_Toc38975571)

[6.14 Получение прикреплений пациента 326](#_Toc38975572)

[6.15 Получение списка изменений прикреплений пациентов за период 327](#_Toc38975573)

[6.16 Получение согласия на обработку персональных данных. 329](#_Toc38975574)

[6.17 Получение данных по сигнальной информации пациента 330](#_Toc38975575)

[6.18 Получение данных по амбулаторной карте по пациенту и МО 334](#_Toc38975576)

[7 Описание методов работы с логами федеральных сервисов 335](#_Toc38975577)

[7.1 Получение лога по человеку и МО 335](#_Toc38975579)

[7.2 Получение лога по паспорту медицинского изделия 335](#_Toc38975580)

[7.3 Получение лога по паспорту МО 335](#_Toc38975581)

[7.4 Получение лога отправки событий (КВС и ТАП) 336](#_Toc38975582)

[8 Описание методов работы со спецификами 337](#_Toc38975583)

[8.1 Получение данных по специфике новорожденного 337](#_Toc38975584)

[8.2 Получение данных по специфике родов 338](#_Toc38975585)

[8.3 Получение данных по специфике психиатрии, наркологии 339](#_Toc38975586)

[8.4 Получение данных по употребляемым психоактивным веществам 340](#_Toc38975587)

[8.5 Создание извещения об инфекционном заболевании (ВИЧ) 341](#_Toc38975588)

[8.6 Получение извещения об инфекционном заболевании (ВИЧ) 341](#_Toc38975589)

[8.7 Создание оперативного донесения о лице, в крови которого при исследовании в реакции иммуноблота выявлены антитела к ВИЧ 342](#_Toc38975590)

[8.8 Получение оперативного донесения о лице, в крови которого при исследовании в реакции иммуноблота выявлены антитела к ВИЧ 343](#_Toc38975591)

[8.9 Создание записи регистра по онкологии, больных туберкулезом и ВИЧ-инфицированых 344](#_Toc38975592)

[8.10 Получение записи регистра по онкологии, больных туберкулезом и ВИЧ-инфицированых 345](#_Toc38975593)

[8.11 Изменение записи регистра по онкологии, больных туберкулезом и ВИЧ-инфицированых 346](#_Toc38975594)

[8.12 Изменение специфики по ВИЧ 347](#_Toc38975595)

[8.13 Получение специфики по ВИЧ 348](#_Toc38975596)

[8.14 Создание химиопрофилактики ВИЧ–инфекции 350](#_Toc38975597)

[8.15 Получение химиопрофилактики ВИЧ–инфекции 350](#_Toc38975598)

[8.16 Изменение химиопрофилактики ВИЧ–инфекции 351](#_Toc38975599)

[8.17 Создание вакцинации в рамках специфики ВИЧ 351](#_Toc38975600)

[8.18 Получение вакцинации в рамках специфики ВИЧ 351](#_Toc38975601)

[8.19 Изменение вакцинации в рамках специфики ВИЧ 352](#_Toc38975602)

[8.20 Создание вторичных заболеваний и оппортунистических инфекций в рамках специфики ВИЧ 352](#_Toc38975603)

[8.21 Получение вторичных заболеваний и оппортунистических инфекций в рамках специфики ВИЧ 352](#_Toc38975604)

[8.22 Изменение вторичных заболеваний и оппортунистических инфекций в рамках специфики ВИЧ 352](#_Toc38975605)

[8.23 Создание извещения о больном туберкулезом 353](#_Toc38975606)

[8.24 Получение извещения о больном туберкулезом 355](#_Toc38975607)

[8.25 Изменение извещения о больном туберкулезом 357](#_Toc38975608)

[8.26 Получение специфики по туберкулезу 360](#_Toc38975609)

[8.27 Изменение специфики по туберкулезу 361](#_Toc38975610)

[8.28 Создание записи Генерализированные формы в рамках специфики по туберкулезу 362](#_Toc38975611)

[8.29 Получение записи Генерализированные формы в рамках специфики по туберкулезу 362](#_Toc38975612)

[8.30 Изменение записи Генерализированные формы в рамках специфики по туберкулезу 363](#_Toc38975613)

[8.31 Создание режима химиотерапии в рамках специфики по туберкулезу 363](#_Toc38975614)

[8.32 Получение режима химиотерапии в рамках специфики по туберкулезу 363](#_Toc38975615)

[8.33 Изменение режима химиотерапии в рамках специфики по туберкулезу 364](#_Toc38975616)

[8.34 Создание консультации фтизиохирурга в рамках специфики по туберкулезу 364](#_Toc38975617)

[8.35 Получение консультации фтизиохирурга в рамках специфики по туберкулезу 364](#_Toc38975618)

[8.36 Изменение консультации фтизиохирурга в рамках специфики по туберкулезу 365](#_Toc38975619)

[8.37 Создание записи об оперативном лечении в рамках специфики по туберкулезу 365](#_Toc38975620)

[8.38 Получение записи об оперативном лечении в рамках специфики по туберкулезу 365](#_Toc38975621)

[8.39 Изменение записи об оперативном лечении в рамках специфики по туберкулезу 365](#_Toc38975622)

[8.40 Создание направления на проведение микроскопических исследований на туберкулез 366](#_Toc38975623)

[8.41 Получение направления на проведение микроскопических исследований на туберкулез 366](#_Toc38975624)

[8.42 Изменение направления на проведение микроскопических исследований на туберкулез 367](#_Toc38975625)

[8.43 Создание результатов микроскопических исследований в рамках специфики по туберкулезу 367](#_Toc38975626)

[8.44 Получение результатов микроскопических исследований в рамках специфики по туберкулезу 368](#_Toc38975627)

[8.45 Изменение результатов микроскопических исследований в рамках специфики по туберкулезу 368](#_Toc38975628)

[8.46 Создание лекарственных назначений в рамках специфики по туберкулезу 368](#_Toc38975629)

[8.47 Получение лекарственных назначений в рамках специфики по туберкулезу 369](#_Toc38975630)

[8.48 Изменение лекарственных назначений в рамках специфики по туберкулезу 369](#_Toc38975631)

[8.49 Создание графика исполнения назначения процедур в рамках специфики по туберкулезу 370](#_Toc38975632)

[8.50 Получение графика исполнения назначения процедур в рамках специфики по туберкулезу 370](#_Toc38975633)

[8.51 Изменение графика исполнения назначения процедур в рамках специфики по туберкулезу 371](#_Toc38975634)

[8.52 Создание результатов исследования в рамках специфики по туберкулезу 371](#_Toc38975635)

[8.53 Получение результатов исследования в рамках специфики по туберкулезу 371](#_Toc38975636)

[8.54 Изменение результатов исследования в рамках специфики по туберкулезу 371](#_Toc38975637)

[8.55 Создание результатов тестов на лекарственную чувствительность в рамках специфики по туберкулезу 372](#_Toc38975638)

[8.56 Получение результатов тестов на лекарственную чувствительность в рамках специфики по туберкулезу 372](#_Toc38975639)

[8.57 Изменение результатов тестов на лекарственную чувствительность в рамках специфики по туберкулезу 373](#_Toc38975640)

[8.58 Создание молекулярно–генетические методов в рамках специфики по туберкулезу 373](#_Toc38975641)

[8.59 Получение молекулярно–генетические методов в рамках специфики по туберкулезу 373](#_Toc38975642)

[8.60 Изменение молекулярно–генетические методов в рамках специфики по туберкулезу 374](#_Toc38975643)

[8.61 Создание микроскопии в рамках специфики по туберкулезу 374](#_Toc38975644)

[8.62 Получение микроскопии в рамках специфики по туберкулезу 374](#_Toc38975645)

[8.63 Изменение микроскопии в рамках специфики по туберкулезу 375](#_Toc38975646)

[8.64 Создание результатов посева в рамках специфики по туберкулезу 375](#_Toc38975647)

[8.65 Получение результатов посева в рамках специфики по туберкулезу 375](#_Toc38975648)

[8.66 Изменение результатов посева в рамках специфики по туберкулезу 376](#_Toc38975649)

[8.67 Создание результатов гистологии в рамках специфики по туберкулезу 376](#_Toc38975650)

[8.68 Получение результатов гистологии в рамках специфики по туберкулезу 376](#_Toc38975651)

[8.69 Изменение результатов гистологии в рамках специфики по туберкулезу 377](#_Toc38975652)

[8.70 Создание результатов рентгена в рамках специфики по туберкулезу 377](#_Toc38975653)

[8.71 Получение результатов рентгена в рамках специфики по туберкулезу 377](#_Toc38975654)

[8.72 Изменение результатов рентгена в рамках специфики по туберкулезу 378](#_Toc38975655)

[8.73 Создание извещения об онкобольном 378](#_Toc38975656)

[8.74 Получение извещения об онкобольном 379](#_Toc38975657)

[8.75 Изменение извещения об онкобольном 381](#_Toc38975658)

[8.76 Создание данных по специфике онкологии 382](#_Toc38975659)

[8.77 Изменение данных по специфике онкологии 384](#_Toc38975660)

[8.78 Получение данных по специфике онкологии 387](#_Toc38975661)

[8.79 Создание данных по химиотерапевтическому лечению в рамках специфики онкологии 388](#_Toc38975662)

[8.80 Изменение данных по химиотерапевтическому лечению в рамках специфики онкологии 389](#_Toc38975663)

[8.81 Получение данных по химиотерапевтическому лечению в рамках специфики онкологии 390](#_Toc38975664)

[8.82 Создание данных по лучевому лечению в рамках специфики онкологии 390](#_Toc38975665)

[8.83 Изменения данных по лучевому лечению в рамках специфики онкологии 391](#_Toc38975666)

[8.84 Получение данных по лучевому лечению в рамках специфики онкологии 392](#_Toc38975667)

[8.85 Создание данных по гормоноиммунотерапевтическому лечению в рамках специфики онкологии 393](#_Toc38975668)

[8.86 Изменения данных по гормоноиммунотерапевтическому лечению в рамках специфики онкологии 394](#_Toc38975669)

[8.87 Получение данных по гормоноиммунотерапевтическому лечению в рамках специфики онкологии 394](#_Toc38975670)

[8.88 Создание данных по хирургическому лечению в рамках специфики онкологии 395](#_Toc38975671)

[8.89 Изменения данных по хирургическому лечению в рамках специфики онкологии 396](#_Toc38975672)

[8.90 Получение данных по хирургическому лечению в рамках специфики онкологии 396](#_Toc38975673)

[8.91 Создание данных по неспецифическому лечению в рамках специфики онкологии 397](#_Toc38975674)

[8.92 Изменение данных по неспецифическому лечению в рамках специфики онкологии 397](#_Toc38975675)

[8.93 Получение данных по неспецифическому лечению в рамках специфики онкологии 397](#_Toc38975676)

[9 Описание методов работы с данными СУР 398](#_Toc38975677)

[9.1 Получение идентификаторов СУР 398](#_Toc38975679)

[10 Описание методов работы с прочими объектами 399](#_Toc38975680)

[10.1 Получение показателей крови человека 399](#_Toc38975682)

[10.2 Получение атрибутов МЭС (КЗГ) 399](#_Toc38975683)

[10.3 Получение данных по амбулаторной карте по пациенту и МО 399](#_Toc38975684)

[10.4 Получение информации по стационарной бирке 400](#_Toc38975685)

[10.5 Получение значениий атрибутов тарифа 400](#_Toc38975686)

[10.6 Получение МО по идентификатору 401](#_Toc38975687)

[10.7 Получение Таблицы связка регионального и ФЕД профиля коек 401](#_Toc38975688)

[11 Описание методов модуля «Имуннопрофилактика» 402](#_Toc38975689)

[Описание методов работы с журналом вакцинации 402](#_Toc38975691)

[11.1 Описание создания записи журнала вакцинации 402](#_Toc38975692)

[11.2 Описание изменения записи журнала вакцинации 403](#_Toc38975693)

[11.3 Описание получения записи журнала вакцинации 404](#_Toc38975694)

[Описание методов работы с туберкулиновой пробой 405](#_Toc38975695)

[11.4 Описание создания записи о туберкулиновой пробе 405](#_Toc38975696)

[11.5 Описание изменения записи о туберкулиновой пробе 406](#_Toc38975697)

[11.6 Описание получения записей о туберкулиновых пробах 407](#_Toc38975698)

[12 Описание методов работы с ресурсами 408](#_Toc38975699)

[12.1 Добавление нового ресурса 408](#_Toc38975700)

[12.2 Редактирование ресурса 408](#_Toc38975701)

[12.3 Добавление расписания на ресурс 409](#_Toc38975702)

[12.4 Изменение расписания на ресурс 409](#_Toc38975703)

[13 Импорт данных из СЭМД «Направление на госпитализацию, восстановительное лечение, обследование, консультацию» (ИЭМК) 411](#_Toc38975704)

[13.1 Схема импорта направлений 411](#_Toc38975705)

ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ

Таблица 1 – Определения и сокращения

|  |  |
| --- | --- |
| **Сокращение** | **Расшифровка сокращения** |
| АРМ | Автоматизированное рабочее место |
| БД | База данных |
| ГОСТ | Государственный стандарт |
| ЕГИСЗ | Единая государственная информационная система в сфере здравоохранения (федеральный фрагмент) |
| ЕСИА | [Единая система идентификации и аутентификации](http://minsvyaz.ru/ru/activity/directions/13/) |
| ЗСПД ЕГИСЗ | Защищенная сеть передачи данных ЕГИСЗ |
| ИНН | Идентификационный номер налогоплательщика |
| ИЭМК | Федеральный сервис Интегрированная электронная медицинская карта |
| КЛАДР | [Классификатор адресов России](http://www.gnivc.ru/inf_provision/classifiers_reference/kladr/) |
| МКБ-10 | Международная классификация болезней 10-го пересмотра |
| МО | Медицинская организация |
| ОМС | Обязательное медицинское страхование |
| ПО | Программное обеспечение |
| РД | Руководящий документ |
| РФ | Российская Федерация |
| СНИЛС | Страховой номер индивидуального лицевого счета гражданина в системе обязательного пенсионного страхования |
| СЭМД | Структурированные электронные медицинские документы |
| ФИО | Фамилия Имя Отчество |
| ФЭР | Федеральный сервис «Федеральная электронная регистратура» единой государственной информационной системы в сфере здравоохранения |
| ЭМК | Электронная медицинская карта пациента |
| API | Application Programming Interface, Интерфейс прикладного программирования - набор методов, предоставляемых сервисом. |
| Call-центр | Специализированная организация или выделенное подразделение в организации, занимающиеся обработкой обращений по записи на прием, поступивших посредством телефонных сетей общего пользования |
| PACS | Система передачи и архивации DICOM изображений |
| REST | Representational State Transfer - «передача состояния представления» - архитектурный стиль взаимодействия компонентов распределённого приложения в сети. |

1. ОБЩАЯ ИНФОРМАЦИЯ
3. 1. Общее описание

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

* + 1. Кэширование запросов

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

* + 1. HTTP ошибки

Общие HTTP ошибки возвращаются HTTP кодами.

401 - не передан или неверный apiKey

404 - обращение к несуществующему методу

* + 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, см. справочник кодов ошибок.
* error\_msg - сообщение c 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.
  + 1. 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);

О – 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],ОM,1@mail.ru) - электронная почта, уникальная
  1. Схема взаимодействия

1. Запрос из сторонней МИС приходит в сервиc.
2. Сервис обрабатывает запрос и вызывает соответствующие методы REST API.
3. Методы REST API производят необходимые манипуляции с данными (поиск, сохранение, удаление).

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

Последовательность вызова REST -методов описана в разделах 1.4- 1.5.

* 1. 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.
  1. Описание последовательности вызова методов при получении запроса на добавление человека

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| **№ п/п** | **Описание метода** | **Метод** | **Передаваемые параметры** | **Комментарий** |
| 1 | Поиск человека | GET api/Person | PersonSurName\_SurName, PersonFirName\_FirName, PersonBirthDay\_BirthDay,  PersonSnils\_Snils | Если не найдено записей, то шаг 2. Если найдена одна запись, то шаг 3. Если найдено несколько записей, то ошибка «Найдено несколько пациентов по указанным параметрам. Просьба уточнить параметры.» и выход. |
| 2 | Создание человека | POST api/Person |  | Если результат выполнения метода успешный, то в ответе возвращаются параметры:  - Person\_id и переход к шагу 4. Иначе ошибка «Ошибка выполнения метода «Создание человека»» и выход. |
| 3 | Редактирование человека | PUT api/Person |  | Если результат выполнения метода успешный, то ответ «Успешно» и переход к шагу 4. Иначе предупреждение «Ошибка выполнения метода «Редактирование человека»» и переход к шагу 4. |
| 4 | Поиск полиса | GET api/Polis | Person\_id, Polis\_Ser, Polis\_Num | Если не найдено записей, то шаг 5. Если найдена одна запись, то шаг 6. Если найдено несколько записей, то предупреждение «Найдено несколько полисов по указанным параметрам. Просьба уточнить параметры.» и переход к шагу 7. |
| 5 | Создание полиса | POST api/Polis |  | Если результат выполнения метода успешный, то в ответе возвращаются параметры:  - Person\_id  - Polis\_id и переход к шагу 4. Иначе предупреждение «Ошибка выполнения метода «Создание полиса»» и переход к шагу 7. |
| 6 | Редактирование полиса | PUT api/Polis |  | Если результат выполнения метода успешный, то ответ «Успешно» и переход к шагу 7. Иначе предупреждение «Ошибка выполнения метода «Редактирование полиса»» и переход к шагу 7. |
| 7 | Поиск документа, удостоверяющего личность | GET api/Document | Person\_id, Document\_Ser, Document\_Num | Если не найдено записей, то шаг 8. Если найдена одна запись, то шаг 9. Если найдено несколько записей, то предупреждение «Найдено несколько документов по указанным параметрам. Просьба уточнить параметры.» и переход к шагу 10. |
| 8 | Создание документа, удостоверяющего личность | POST api/Document |  | Если результат выполнения метода успешный, то в ответе возвращаются параметры:  - Person\_id - Document\_id и переход к шагу 10. Иначе предупреждение «Ошибка выполнения метода «Создание документа, удостоверяющего личность»» и переход к шагу 10. |
| 9 | Редактирование документа, удостовреяющего личность | PUT api/Document |  | Если результат выполнения метода успешный, то ответ «Успешно» и переход к шагу 10. Иначе предупреждение «Ошибка выполнения метода «Редактирование документа, удостворяющего личность»» и переход к шагу 10. |
| 10 | Поиск адреса регистрации | GET api/Address | Person\_id, AddressType\_id = 1 | Если не найдено записей, то шаг 11. Если найдена одна запись, то шаг 12. Если найдено несколько записей, то предупреждение «Найдено несколько адресов по указанным параметрам. Просьба уточнить параметры.» и переход к шагу 13. |
| 11 | Создание адреса регистрации | POST api/Address |  | Если результат выполнения метода успешный, то в ответе возвращаются параметры:  • Person\_id (N, Н) – Идентификатор человека; • Address\_id (N, О) – Идентификатор адреса; • AddressType\_id (N, Н) – Тип адреса (1 - Адрес регистрации, 2 - Адрес проживания, 3 - Адрес рождения) и переход к шагу 13. Иначе предупреждение «Ошибка выполнения метода «Создание адреса регистрации»» и переход к шагу 13. |
| 12 | Редактирование адреса регистрации | PUT api/Address |  | Если результат выполнения метода успешный, то ответ «Успешно» и переход к шагу 13. Иначе предупреждение «Ошибка выполнения метода «Редактирование Адреса регистрации»» и переход к шагу 13. |
| 13 | Поиск адреса проживания | GET api/Address | Person\_id, AddressType\_id = 2 | Если не найдено записей, то шаг 14. Если найдена одна запись, то шаг 15. Если найдено несколько записей, то предупреждение «Найдено несколько адресов по указанным параметрам. Просьба уточнить параметры.» и переход к шагу 16. |
| 14 | Создание адреса проживания | POST api/Address |  | Если результат выполнения метода успешный, то в ответе возвращаются параметры:  • Person\_id (N, Н) – Идентификатор человека; • Address\_id (N, О) – Идентификатор адреса; • AddressType\_id (N, Н) – Тип адреса (1 - Адрес регистрации, 2 - Адрес проживания, 3 - Адрес рождения) и переход к шагу 16. Иначе предупреждение «Ошибка выполнения метода «Создание адреса проживания»» и переход к шагу 16. |
| 15 | Редактирование адреса проживания | PUT api/Address |  | Если результат выполнения метода успешный, то ответ «Успешно» и переход к шагу 16. Иначе предупреждение «Ошибка выполнения метода «Редактирование Адреса проживания»» и переход к шагу 16. |
| 16 | Поиск адреса рождения | GET api/Address | Person\_id, AddressType\_id = 3 | Если не найдено записей, то шаг 17. Если найдена одна запись, то шаг 18. Если найдено несколько записей, то предупреждение «Найдено несколько адресов по указанным параметрам. Просьба уточнить параметры.» и выход. |
| 17 | Создание адреса рождения | POST api/Address |  | Если результат выполнения метода успешный, то в ответе возвращаются параметры:  • Person\_id (N, Н) – Идентификатор человека; • Address\_id (N, О) – Идентификатор адреса; • AddressType\_id (N, Н) – Тип адреса (1 - Адрес регистрации, 2 - Адрес проживания, 3 - Адрес рождения) и выход. Иначе предупреждение «Ошибка выполнения метода «Создание адреса рождения»» и выход. |
| 18 | Редактирование адреса рождения | PUT api/Address |  | Если результат выполнения метода успешный, то ответ «Успешно» и выход. Иначе предупреждение «Ошибка выполнения метода «Редактирование Адреса рождения»» и выход. |

* 1. Описание последовательности вызова методов при получении документа

Общее описание и описание формата передачи данных приведено в разделе 4.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Входной документ: "Эпикриз в амбулатории"** | | | | |
| **№ п/п** | **Описание метода** | **Метод** | **Передаваемые параметры** | **Комментарий** |
| 1 | Поиск человека | GET api/Person | Person\_id, PersonSurName\_SurName, PersonFirName\_FirName, PersonBirthDay\_BirthDay, PersonSnils\_Snils | Если передается Person\_id, то вызов метода с параметром Person\_id. Если запись не найдена, то параметризированный поиск. Если Person\_id не передается или не найден в БД, то вызов метода с набором параметров пока не будет найдена хотя бы одна запись (вызов метода с первым набором параметров, если запись найдена, то шаг 3, если найдено несколько записей, то шаг 2, если запись не найдена, то вызов метода со следующим набором параметров и т.д.). Наборы параметров : •PersonSurName\_SurName, PersonFirName\_FirName, PersonBirthDay\_BirthDay, PersonSnils\_Snils •PersonSurName\_SurName, PersonFirName\_FirName, PersonBirthDay\_BirthDay •PersonSurName\_SurName, PersonFirName\_FirName, PersonSnils\_Snils Если не найдено записей или найдено несколько записей, то шаг 2. Если найдена одна запись, то шаг 3. |
| 2 | Создание человека | POST api/Person |  | В ответе возвращать параметры:  - Person\_id |
| 3 | Редактирование человека | PUT api/Person |  |  |
| 4 | Поиск полиса | GET api/Polis | Person\_id, Polis\_Ser, Polis\_Num | Если не найдено записей, то шаг 5. Если найдена одна запись, то шаг 6. Если найдено несколько записей, то ошибка. |
| 5 | Создание полиса | POST api/Polis |  | В ответе возвращать параметры:  - Person\_id  - Polis\_id |
| 6 | Редактирование полиса | PUT api/Polis |  |  |
| 7 | Поиск документа, удостоверяющего личность | GET api/Document | Person\_id, Document\_Ser, Document\_Num | Если не найдено записей, то шаг 8. Если найдена одна запись, то шаг 9. Если найдено несколько записей, то ошибка. |
| 8 | Создание документа, удостоверяющего личность | POST api/Document |  | В ответе возвращать параметры:  - Person\_id - Document\_id |
| 9 | Редактирование документа, удостовреяющего личность | PUT api/Document |  |  |
| 10 | Поиск адреса регистрации | GET api/Address | Person\_id, AddressType\_id = 1 | Если не найдено записей, то шаг 11. Если найдена одна запись, то шаг 12. Если найдено несколько записей, то ошибка. |
| 11 | Создание адреса регистрации | POST api/Address |  | В ответе возвращать параметры:  • Person\_id (N, Н) – Идентификатор человека; • Address\_id (N, О) – Идентификатор адреса; • AddressType\_id (N, Н) – Тип адреса (1 - Адрес регистрации, 2 - Адрес проживания, 3 - Адрес рождения) |
| 12 | Редактирование адреса регистрации | PUT api/Address |  |  |
| 13 | Поиск адреса проживания | GET api/Address | Person\_id, AddressType\_id = 2 | Если не найдено записей, то шаг 14. Если найдена одна запись, то шаг 15. Если найдено несколько записей, то ошибка. |
| 14 | Создание адреса проживания | POST api/Address |  | В ответе возвращать параметры:  • Person\_id (N, Н) – Идентификатор человека; • Address\_id (N, О) – Идентификатор адреса; • AddressType\_id (N, Н) – Тип адреса (1 - Адрес регистрации, 2 - Адрес проживания, 3 - Адрес рождения) |
| 15 | Редактирование адреса проживания | PUT api/Address |  |  |
| 16 | Поиск адреса рождения | GET api/Address | Person\_id, AddressType\_id = 3 | Если не найдено записей, то шаг 17. Если найдена одна запись, то шаг 18. Если найдено несколько записей, то ошибка. |
| 17 | Создание адреса рождения | POST api/Address |  | В ответе возвращать параметры:  • Person\_id (N, Н) – Идентификатор человека; • Address\_id (N, О) – Идентификатор адреса; • AddressType\_id (N, Н) – Тип адреса (1 - Адрес регистрации, 2 - Адрес проживания, 3 - Адрес рождения) |
| 18 | Редактирование адреса рождения | PUT api/Address |  |  |
| 19 | Поиск ТАП | GET api/EvnPLBase | Person\_id, EvnPL\_NumCard | Если не найдено записей, то шаг 20. Если найдена одна запись, то шаг 21. Если найдено несколько записей, то ошибка. |
| 20 | Создание ТАП | POST api/EvnPLBase |  | В ответе возвращать параметры:  • EvnPLBase\_id (N,О) – Идентификатор ТАП |
| 21 | Редактирование ТАП | PUT api/EvnPLBase |  |  |
| 22 | Поиск посещения | GET api/EvnVizitPL | EvnPLBase\_id, EvnVizitPL\_Date | Если не найдено записей, то шаг 23. Если найдена одна запись, то шаг 24. Если найдено несколько записей, то ошибка. |
| 23 | Создание посещения | POST api/EvnVizitPL |  | В ответе возвращать параметры:  • EvnVizit\_id (N, О) - Идентификатор посещения |
| 24 | Редактирование посещения | PUT api/EvnVizitPL |  |  |
| 25 | Поиск общей услуги | GET api/EvnUsluga | Evn\_pid, Evn\_setDT, UslugaComplex\_id | Если не найдено записей, то шаг 26. Если найдена одна запись, то шаг 27. Если найдено несколько записей, то ошибка. |
| 26 | Создание общей услуги | POST api/EvnUsluga |  | В ответе возвращать параметры:  • EvnUsluga\_id (N, О) – Идентификатор оказания услуги; • Evn\_id (N, О) – Идентификатор случая оказания услуги (значение сущности dbo.Evn) |
| 27 | Редактирование общей услуги | PUT api/EvnUsluga |  |  |
| 28 | Поиск оперативной услуги | GET api/EvnUslugaOper | Evn\_pid, Evn\_setDT, UslugaComplex\_id | Если не найдено записей, то шаг 29. Если найдена одна запись, то шаг 30. Если найдено несколько записей, то ошибка. |
| 29 | Создание оперативной услуги | POST api/EvnUslugaOper |  | В ответе возвращать параметры:  • EvnUsluga\_id (N, О) – Идентификатор оказания услуги; • EvnUslugaOper\_id (N, О) – Идентификатор оказания оперативной услуги; • Evn\_id (N, О) – Идентификатор случая оказания услуги (значение сущности dbo.Evn) |
| 30 | Редактирование оперативной услуги | PUT api/EvnUslugaOper |  |  |
| 31 | Поиск анестезии | GET api/EvnUslugaOperAnest | EvnUslugaOper\_id, AnesthesiaClass\_id | Если не найдено записей, то шаг 32. Если найдена одна запись, то шаг 33. Если найдено несколько записей, то ошибка. |
| 32 | Создание анестезии на услуге | POST api/EvnUslugaOperAnest |  | В ответе возвращать параметры:  • EvnUslugaOperAnest\_id (N, О) – Идентификатор использования анестезии |
| 33 | Редактирование анестезии | PUT api/EvnUslugaOperAnest |  |  |
| 34 | Поиск осложнения | GET api/EvnAgg | Evn\_id, AggType\_id | Если не найдено записей, то шаг 35. Если найдена одна запись, то шаг 36. Если найдено несколько записей, то ошибка. |
| 35 | Создание осложнения на услуге | POST api/EvnAgg |  | В ответе возвращать параметры:  • EvnAgg\_id (N, О) – Идентификатор осложнения |
| 36 | Редактирование осложнения | PUT api/EvnAgg |  |  |
| 37 | Поиск ЛВН | GET api/Stick | Evn\_pid, EvnStick\_num, EvnStick\_SetDate | Если не найдено записей, то шаг 38. Если найдена одна запись, то шаг 39. Если найдено несколько записей, то ошибка. |
| 38 | Создание ЛВН | POST api/Stick |  | В ответе возвращать параметры:  • EvnStickBase\_id (N, О) – Идентификатор ЛВН; • Evn\_id (N, О) – Идентификатор события выписки документа нетрудоспособности |
| 39 | Редактирование ЛВН | PUT api/Stick |  |  |
| 40 | Поиск справки учащегося | GET api/StickStudent | Evn\_pid, EvnStick\_num, EvnStick\_SetDate | Если не найдено записей, то шаг 41. Если найдена одна запись, то шаг 42. Если найдено несколько записей, то ошибка. |
| 41 | Создание справки учащегося | POST api/StickStudent |  | В ответе возвращать параметры:  • EvnStickBase\_id (N, О) – Идентификатор ЛВН; • Evn\_id (N, О) – Идентификатор события выписки документа нетрудоспособности |
| 42 | Редактирование справки учащегося | PUT api/StickStudent |  |  |
| 43 | Поиск периода освобождения от работы | GET api/WorkRelease | EvnStickBase\_id, EvnStickWorkRelease\_begDate, EvnStickWorkRelease\_endDate | Если не найдено записей, то шаг 44. Если найдена одна запись, то шаг 45. Если найдено несколько записей, то ошибка. |
| 44 | Создание периода освобождения от работы | POST api/WorkRelease |  | В ответе возвращать параметры:  • EvnStickBase\_id (N, О) – Идентификатор документа нетрудоспособности; • EvnStickWorkRelease\_id (N, О) – идентификатор периода освобождения от работы |
| 45 | Редактирование периода освобождения от работы | PUT api/WorkRelease |  |  |
| 46 | Поиск льготного рецепта | GET api/EvnRecept | Evn\_pid, Evn\_setDT, EvnRecept\_Num | Если не найдено записей, то шаг 47. Если найдена одна запись, то шаг 48. Если найдено несколько записей, то ошибка. |
| 47 | Создание льготного рецепта | POST api/EvnRecept |  | В ответе возвращать параметры:  • Evn\_id (N, О) – Идентификатор случая выписки льготного рецепта |
| 48 | Редактирование льготного рецепта | PUT api/EvnRecept |  |  |
| 49 | Поиск прочего документа - Первичного осмотра | GET api/EvnXml | Evn\_id, XmlType\_id = "3. Протокол осмотра" | Если не найдено записей, то шаг 50. Если найдена одна запись, то шаг 51. Если найдено несколько записей, то ошибка. |
| 50 | Создание прочего документа - Первичного осмотра | POST api/EvnXml |  | В ответе возвращать параметры:  • Evn\_id (N, О) – идентификатор случая; • EvnXml\_id (N, О) – идентификатор документа |
| 51 | Редактирование прочего документа - Первичного осмотра | PUT api/EvnXml |  |  |
| 52 | Поиск прочего документа - Эпикриза | GET api/EvnXml | Evn\_id, XmlType\_id = "2. Документ в свободной форме" | Если не найдено записей, то шаг 53. Если найдена одна запись, то шаг 54. Если найдено несколько записей, то ошибка. |
| 53 | Создание прочего документа - Эпикриза | POST api/EvnXml |  | В ответе возвращать параметры:  • Evn\_id (N, О) – идентификатор случая; • EvnXml\_id (N, О) – идентификатор документа |
| 54 | Редактирование прочего документа - Эпикриза | PUT api/EvnXml |  |  |
| 55 | Поиск назначения лекарственных средств | GET api/EvnPrescrTreat | Evn\_pid, EvnClass\_id = 63, Evn\_setDT, DrugComplexMnn\_id | Если не найдено записей, то шаг 56. Если найдена одна запись, то шаг 57. Если найдено несколько записей, то ошибка. |
| 56 | Создание назначения лекарственных средств | POST api/EvnPrescrTreat |  | В ответе возвращать параметры:  • EvnPrescr\_id (N, О) – Идентификатор назначения; • EvnPrescrTreat\_id (N, О) – Идентификатор назначения лекарственного средства; • Evn\_id (N, О) – Идентификатор события; • EvnCourse\_id (N, О) – Идентификатор курса; • EvnCourseTreat\_id (N, О) – Идентификатор курса приема лекарственных средств |
| 57 | Редактирование назначения лекарственных средств | PUT api/EvnPrescrTreat |  |  |
|  |  |  |  |  |
|  | **Входной документ: "Эпикриз в стационаре выписной"** | | |  |
| **№ п/п** | **Описание метода** | **Метод** | **Как вызывается** | **Комментарий** |
| 1 | Поиск человека | GET api/Person | Person\_id, PersonSurName\_SurName, PersonFirName\_FirName, PersonBirthDay\_BirthDay,  PersonSnils\_Snils | Если передается Person\_id, то вызов метода с параметром Person\_id. Если запись не найдена, то параметризированный поиск. Если Person\_id не передается или не найден в БД, то вызов метода с набором параметров пока не будет найдена хотя бы одна запись (вызов метода с первым набором параметров, если запись найдена, то шаг 3, если найдено несколько записей, то шаг 2, если запись не найдена, то вызов метода со следующим набором параметров и т.д.). Наборы параметров: •PersonSurName\_SurName, PersonFirName\_FirName, PersonBirthDay\_BirthDay, PersonSnils\_Snils •PersonSurName\_SurName, PersonFirName\_FirName, PersonBirthDay\_BirthDay •PersonSurName\_SurName, PersonFirName\_FirName, PersonSnils\_Snils Если не найдено записей или найдено несколько записей, то шаг 2. Если найдена одна запись, то шаг 3. |
| 2 | Создание человека | POST api/Person |  | В ответе возвращать параметры:  - Person\_id |
| 3 | Редактирование человека | PUT api/Person |  |  |
| 4 | Поиск полиса | GET api/Polis | Person\_id, Polis\_Ser, Polis\_Num | Если не найдено записей, то шаг 5. Если найдена одна запись, то шаг 6. Если найдено несколько записей, то ошибка. |
| 5 | Создание полиса | POST api/Polis |  | В ответе возвращать параметры:  - Person\_id  - Polis\_id |
| 6 | Редактирование полиса | PUT api/Polis |  |  |
| 7 | Поиск документа, удостоверяющего личность | GET api/Document | Person\_id, Document\_Ser, Document\_Num | Если не найдено записей, то шаг 8. Если найдена одна запись, то шаг 9. Если найдено несколько записей, то ошибка. |
| 8 | Создание документа, удостоверяющего личность | POST api/Document |  | В ответе возвращать параметры:  - Person\_id - Document\_id |
| 9 | Редактирование документа, удостовреяющего личность | PUT api/Document |  |  |
| 10 | Поиск адреса регистрации | GET api/Address | Person\_id, AddressType\_id = 1 | Если не найдено записей, то шаг 11. Если найдена одна запись, то шаг 12. Если найдено несколько записей, то ошибка. |
| 11 | Создание адреса регистрации | POST api/Address |  | В ответе возвращать параметры:  • Person\_id (N, Н) – Идентификатор человека; • Address\_id (N, О) – Идентификатор адреса; • AddressType\_id (N, Н) – Тип адреса (1 - Адрес регистрации, 2 - Адрес проживания, 3 - Адрес рождения) |
| 12 | Редактирование адреса регистрации | PUT api/Address |  |  |
| 13 | Поиск адреса проживания | GET api/Address | Person\_id, AddressType\_id = 2 | Если не найдено записей, то шаг 14. Если найдена одна запись, то шаг 15. Если найдено несколько записей, то ошибка. |
| 14 | Создание адреса проживания | POST api/Address |  | В ответе возвращать параметры:  • Person\_id (N, Н) – Идентификатор человека; • Address\_id (N, О) – Идентификатор адреса; • AddressType\_id (N, Н) – Тип адреса (1 - Адрес регистрации, 2 - Адрес проживания, 3 - Адрес рождения) |
| 15 | Редактирование адреса проживания | PUT api/Address |  |  |
| 16 | Поиск адреса рождения | GET api/Address | Person\_id, AddressType\_id = 3 | Если не найдено записей, то шаг 17. Если найдена одна запись, то шаг 18. Если найдено несколько записей, то ошибка. |
| 17 | Создание адреса рождения | POST api/Address |  | В ответе возвращать параметры:  • Person\_id (N, Н) – Идентификатор человека; • Address\_id (N, О) – Идентификатор адреса; • AddressType\_id (N, Н) – Тип адреса (1 - Адрес регистрации, 2 - Адрес проживания, 3 - Адрес рождения) |
| 18 | Редактирование адреса рождения | PUT api/Address |  |  |
| 19 | Поиск КВС | GET api/EvnPS | Person\_id, EvnPS\_NumCard, EvnPS\_setDate | Если не найдено записей, то шаг 20. Если найдена одна запись, то шаг 21. Если найдено несколько записей, то ошибка. |
| 20 | Создание КВС | POST api/EvnPS |  | В ответе возвращать параметры:  • EvnPS\_id (N,O) Идентификатор КВС |
| 21 | Редактирование КВС | PUT api/EvnPS |  |  |
| 22 | Поиск движения | GET api/EvnSection | EvnPS\_id, EvnSection\_setDate | Если не найдено записей, то шаг 23. Если найдена одна запись, то шаг 24. Если найдено несколько записей, то ошибка. |
| 23 | Создание движения | POST api/EvnSection |  | В ответе возвращать параметры:  • EvnSection\_id (N,O) идентификатор движения |
| 24 | Редактирование движения | PUT api/EvnSection |  |  |
| 25 | Поиск общей услуги | GET api/EvnUsluga | Evn\_pid, Evn\_setDT, UslugaComplex\_id | Если не найдено записей, то шаг 26. Если найдена одна запись, то шаг 27. Если найдено несколько записей, то ошибка. |
| 26 | Создание общей услуги | POST api/EvnUsluga |  | В ответе возвращать параметры:  • EvnUsluga\_id (N, О) – Идентификатор оказания услуги; • Evn\_id (N, О) – Идентификатор случая оказания услуги (значение сущности dbo.Evn) |
| 27 | Редактирование общей услуги | PUT api/EvnUsluga |  |  |
| 28 | Поиск оперативной услуги | GET api/EvnUslugaOper | Evn\_pid, Evn\_setDT, UslugaComplex\_id | Если не найдено записей, то шаг 29. Если найдена одна запись, то шаг 30. Если найдено несколько записей, то ошибка. |
| 29 | Создание оперативной услуги | POST api/EvnUslugaOper |  | В ответе возвращать параметры:  • EvnUsluga\_id (N, О) – Идентификатор оказания услуги; • EvnUslugaOper\_id (N, О) – Идентификатор оказания оперативной услуги; • Evn\_id (N, О) – Идентификатор случая оказания услуги (значение сущности dbo.Evn) |
| 30 | Редактирование оперативной услуги | PUT api/EvnUslugaOper |  |  |
| 31 | Поиск анестезии | GET api/EvnUslugaOperAnest | EvnUslugaOper\_id, AnesthesiaClass\_id | Если не найдено записей, то шаг 32. Если найдена одна запись, то шаг 33. Если найдено несколько записей, то ошибка. |
| 32 | Создание анестезии на услуге | POST api/EvnUslugaOperAnest |  | В ответе возвращать параметры:  • EvnUslugaOperAnest\_id (N, О) – Идентификатор использования анестезии |
| 33 | Редактирование анестезии | PUT api/EvnUslugaOperAnest |  |  |
| 34 | Поиск осложнения | GET api/EvnAgg | Evn\_id, AggType\_id | Если не найдено записей, то шаг 35. Если найдена одна запись, то шаг 36. Если найдено несколько записей, то ошибка. |
| 35 | Создание осложнения на услуге | POST api/EvnAgg |  | В ответе возвращать параметры:  • EvnAgg\_id (N, О) – Идентификатор осложнения |
| 36 | Редактирование осложнения | PUT api/EvnAgg |  |  |
| 37 | Поиск ЛВН | GET api/Stick | Evn\_pid, EvnStick\_num, EvnStick\_SetDate | Если не найдено записей, то шаг 38. Если найдена одна запись, то шаг 39. Если найдено несколько записей, то ошибка. |
| 38 | Создание ЛВН | POST api/Stick |  | В ответе возвращать параметры:  • EvnStickBase\_id (N, О) – Идентификатор ЛВН; • Evn\_id (N, О) – Идентификатор события выписки документа нетрудоспособности |
| 39 | Редактирование ЛВН | PUT api/Stick |  |  |
| 40 | Поиск справки учащегося | GET api/StickStudent | Evn\_pid, EvnStick\_num, EvnStick\_SetDate | Если не найдено записей, то шаг 41. Если найдена одна запись, то шаг 42. Если найдено несколько записей, то ошибка. |
| 41 | Создание справки учащегося | POST api/StickStudent |  | В ответе возвращать параметры:  • EvnStickBase\_id (N, О) – Идентификатор ЛВН; • Evn\_id (N, О) – Идентификатор события выписки документа нетрудоспособности |
| 42 | Редактирование справки учащегося | PUT api/StickStudent |  |  |
| 43 | Поиск периода освобождения от работы | GET api/WorkRelease | EvnStickBase\_id, EvnStickWorkRelease\_begDate, EvnStickWorkRelease\_endDate | Если не найдено записей, то шаг 44. Если найдена одна запись, то шаг 45. Если найдено несколько записей, то ошибка. |
| 44 | Создание периода освобождения от работы | POST api/WorkRelease |  | В ответе возвращать параметры:  • EvnStickBase\_id (N, О) – Идентификатор документа нетрудоспособности; • EvnStickWorkRelease\_id (N, О) – идентификатор периода освобождения от работы |
| 45 | Редактирование периода освобождения от работы | PUT api/WorkRelease |  |  |
| 46 | Поиск льготного рецепта | GET api/EvnRecept | Evn\_pid, Evn\_setDT, EvnRecept\_Num | Если не найдено записей, то шаг 47. Если найдена одна запись, то шаг 48. Если найдено несколько записей, то ошибка. |
| 47 | Создание льготного рецепта | POST api/EvnRecept |  | В ответе возвращать параметры:  • Evn\_id (N, О) – Идентификатор случая выписки льготного рецепта |
| 48 | Редактирование льготного рецепта | PUT api/EvnRecept |  |  |
| 49 | Поиск прочего документа - Первичного осмотра | GET api/EvnXml | Evn\_id, XmlType\_id = "8. Осмотр в стационаре" | Если не найдено записей, то шаг 50. Если найдена одна запись, то шаг 51. Если найдено несколько записей, то ошибка. |
| 50 | Создание прочего документа - Первичного осмотра | POST api/EvnXml |  | В ответе возвращать параметры:  • Evn\_id (N, О) – идентификатор случая; • EvnXml\_id (N, О) – идентификатор документа |
| 51 | Редактирвоание прочего документа - Первичного осмотра | PUT api/EvnXml |  |  |
| 52 | Поиск прочего документа - Эпикриза | GET api/EvnXml | Evn\_id, XmlType\_id = "10. Эпикриз" | Если не найдено записей, то шаг 53. Если найдена одна запись, то шаг 54. Если найдено несколько записей, то ошибка. |
| 53 | Создание прочего документа - Эпикриза | POST api/EvnXml |  | В ответе возвращать параметры:  • Evn\_id (N, О) – идентификатор случая; • EvnXml\_id (N, О) – идентификатор документа |
| 54 | Редактирование прочего документа - Эпикриза | PUT api/EvnXml |  |  |
| 55 | Поиск назначения лекарственных средств | GET api/EvnPrescrTreat | Evn\_pid, EvnClass\_id = 63, Evn\_setDT, DrugComplexMnn\_id | Если не найдено записей, то шаг 56. Если найдена одна запись, то шаг 57. Если найдено несколько записей, то ошибка. |
| 56 | Создание назначения лекарственных средств | POST api/EvnPrescrTreat |  | В ответе возвращать параметры:  • EvnPrescr\_id (N, О) – Идентификатор назначения; • EvnPrescrTreat\_id (N, О) – Идентификатор назначения лекарственного средства; • Evn\_id (N, О) – Идентификатор события; • EvnCourse\_id (N, О) – Идентификатор курса; • EvnCourseTreat\_id (N, О) – Идентификатор курса приема лекарственных средств |
| 57 | Редактирование назначения лекарственных средств | PUT api/EvnPrescrTreat |  |  |
|  |  |  |  |  |
| **Входной документ: "Направление на консультацию и вспомогательные кабинеты"** | | | | |
| **№ п/п** | **Описание метода** | **Метод** | **Передаваемые параметры** | **Комментарий** |
| 1 | Поиск человека | GET api/Person | Person\_id, PersonSurName\_SurName, PersonFirName\_FirName, PersonBirthDay\_BirthDay,  PersonSnils\_Snils | Если передается Person\_id, то вызов метода с параметром Person\_id. Если запись не найдена, то параметризированный поиск. Если Person\_id не передается или не найден в БД, то вызов метода с набором параметров пока не будет найдена хотя бы одна запись (вызов метода с первым набором параметров, если запись найдена, то шаг 3, если найдено несколько записей, то шаг 2, если запись не найдена, то вызов метода со следующим набором параметров и т.д.). Наборы параметров : •PersonSurName\_SurName , PersonFirName\_FirName, PersonBirthDay\_BirthDay, PersonSnils\_Snils •PersonSurName\_SurName , PersonFirName\_FirName, PersonBirthDay\_BirthDay •PersonSurName\_SurName , PersonFirName\_FirName, PersonSnils\_Snils Если не найдено записей или найдено несколько записей, то шаг 2. Если найдена одна запись, то шаг 3. |
| 2 | Создание человека | POST api/Person |  |  |
| 3 | Редактирование человека | PUT api/Person |  |  |
| 4 | Поиск полиса | GET api/Polis | Person\_id, Polis\_Ser, Polis\_Num | Если не найдено записей, то шаг 5. Если найдена одна запись, то шаг 6. Если найдено несколько записей, то ошибка. |
| 5 | Создание полиса | POST api/Polis |  |  |
| 6 | Редактирование полиса | PUT api/Polis |  |  |
| 7 | Поиск документа, удостоверяющего личность | GET api/Document | Person\_id, Document\_Ser, Document\_Num | Если не найдено записей, то шаг 8. Если найдена одна запись, то шаг 9. Если найдено несколько записей, то ошибка. |
| 8 | Создание документа, удостоверяющего личность | POST api/Document |  |  |
| 9 | Редактирование документа, удостовреяющего личность | PUT api/Document |  |  |
| 10 | Поиск адреса регистрации | GET api/Address | Person\_id, AddressType\_id = 1 | Если не найдено записей, то шаг 11. Если найдена одна запись, то шаг 12. Если найдено несколько записей, то ошибка. |
| 11 | Создание адреса регистрации | POST api/Address |  |  |
| 12 | Редактирование адреса регистрации | PUT api/Address |  |  |
| 13 | Поиск адреса проживания | GET api/Address | Person\_id, AddressType\_id = 2 | Если не найдено записей, то шаг 14. Если найдена одна запись, то шаг 15. Если найдено несколько записей, то ошибка. |
| 14 | Создание адреса проживания | POST api/Address |  |  |
| 15 | Редактирование адреса проживания | PUT api/Address |  |  |
| 16 | Поиск адреса рождения | GET api/Address | Person\_id, AddressType\_id = 3 | Если не найдено записей, то шаг 17. Если найдена одна запись, то шаг 18. Если найдено несколько записей, то ошибка. |
| 17 | Создание адреса рождения | POST api/Address |  |  |
| 18 | Редактирование адреса рождения | PUT api/Address |  |  |
|  |  |  |  |  |

* 1. Сервис извещений об изменениях в расписании, записи на прием, очереди

Сервис извещений об изменениях в расписании, записи на прием, очереди (далее «Сервис извещений об изменениях»).

Сервис выполняет следующие функции:

* Загрузка данных об изменениях в отдельное хранилище.
* Хранение данных об изменениях в отдельном хранилище.
* Оперативная передача данных об изменениях сторонним МИС.
  + 1. Схема взаимодействия

Загрузка данных в отдельное хранилище производится автоматически по следующим событиям:

* Изменения по биркам поликлиники:
  + Создание бирки;
  + Изменения по бирке (в том числе запись на бирку);
  + Удаление бирки.
* Изменения в очереди на прием к врачу поликлиники (постановка в очередь):
  + Создание постановки пациента в очередь;
  + Изменение очереди;
  + Удаление очереди.

Загрузка данных в отдельное хранилище производится сразу после сохранения данных в БД Системы.

**Хранение данных об изменениях в отдельном хранилище**:

Данные об изменениях хранятся в отдельном хранилище в течение времени, заданного в настройках конфигурационного файла. По истечении указанного времени данные об изменениях автоматически по заданию удаляются из хранилища.

**Передача данных об изменениях сторонним МИС**:

Передача данных об изменениях сторонним МИС производится по REST-запросам. При обработке поступившего REST-запроса происходит обращение НЕ к БД Системы, а к хранилищу данных с изменениями.

* + 1. Используемые методы
       1. Получение данных об изменениях по биркам поликлиники

**GET api/TimeTableGraf/TimeTableGrafByUpdPeriod**

При выполнении метода обращение производится НЕ к БД Системы, а к отдельному хранилищу данных с изменениями.

Входящие параметры:

* Lpu\_id (N, Н) – МО;
* TimeTableGraf\_updbeg (D, О) – Дата начала периода изменений;
* TimeTableGraf\_updend (D, О) – Дата окончания периода изменений.  
  Ответ:
* Проверка: период между TimeTableGraf\_updbeg и TimeTableGraf\_updend НЕ должен превышать 1 мес.
* Успешный ответ – список бирок и записей на прием к врачу, отфильтрованный по дате изменений в БД (TimeTableGraf\_updDT between TimeTableGraf\_updbeg and TimeTableGraf\_updend):
  + Lpu\_id (N, О) – МО;
  + MedStaffFact\_id (N, О) – Место работы врача;
  + TimeTableGraf\_id (N, O) - Идентификатор бирки;
  + TimeTableGraf\_begTime (DT, О) – Дата и время приема;
  + TimeTableType\_id (N, Н) – Тип бирки.
  + Person\_id (N, O) – Идентификатор пациента;
  + TimeTableGrafAction\_id (N, О) – Ид последней операции с биркой;
  + TimeTableGraf\_insDT (D, О) – Дата создания записи в БД;
  + TimeTableGraf\_updDT (D, О) – Дата изменения записи в БД.
    - 1. Получение данных об изменениях в очереди на прием к врачу поликлиники

**GET api/EvnQueue/EvnQueueByUpdPeriod**

При выполнении метода обращение производится НЕ к БД Системы, а к отдельному хранилищу данных с изменениями.

Входящие параметры:

* Lpu\_did (N, Н) – МО;
* Evn\_updbeg (D, О) – Дата начала периода изменений;
* Evn\_updend (D, О) – Дата окончания периода изменений.  
  Ответ:
* Проверка: период между Evn\_updbeg и Evn\_updend не превышает 1 месяц.
* Успешный ответ – список пациентов, поставленных в очередь, отфильтрованный по дате изменений в БД (Evn\_updDT between Evn\_updbeg and Evn\_updend).
  + Lpu\_did (N, О) – МО, куда поставлен в очередь;
  + LpuUnit\_did (N, О) – Группа отделений, куда поставлен в очередь;
  + LpuSection\_did (N, О) – Отделение МО, куда поставлен в очередь;
  + LpuSectionProfile\_did (N, О) – Профиль отделения МО, куда поставлен в очередь;
  + MedPersonal\_did (N, О) – Сотрудник, к которому поставлен в очередь;
  + EvnQueue\_id (N, O) – Ид постановки в очередь;
  + Person\_id (N, О) – Ид человека;
  + Evn\_deleted (N, Н) – Признак «Удалено»;
  + Evn\_insDT (D, О) – Дата создания записи в БД
  + Evn\_updDT (D, О) – Дата изменения записи в БД.

1. Системные методы
3. 1. Общие системные методы
4. 1. 1. Авторизация в системе

**GET /api/user/login**

Авторизация пользователя в системе

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

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

Авторизация в РИШ зависит от используемых механизмов:

1. 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 параметров является идентификатор сессии.

1. 2. При использовании SOAP запросов применяется метод WS-security:
   * В секцию <wsse:UsernameToken> всегда передается пара логина и пароля.
     1. Выход из системы

**GET api/user/logout**

Выход из системы

|  |  |
| --- | --- |
| **Входящие параметры** | * Sess\_id (T [], О) – идентификатор сессии |
| **Ответ** | Успешный ответ – ошибка с кодом 0 |
| **Пример** | Запрос  http://<ip>/api/user/logout |

* + 1. Фильтрация данных и доступ к редактированию данных МО

После успешной авторизации в системе пользователь должен установить МО редактирования и просмотра данных методом **PUT api/user/ReviewMO**.

При выполнении дальнейших запросов происходит фильтрация результатов (в случае выполнения запросов типа GET) по МО просмотра, при этом параметр LPU\_id необходимо получать из сессии.

При изменении данных (выполнение запросов типа POST, PUT, DELETE) при отсутствии параметра LPU\_id в методе, все действия производятся, также, в МО просмотра.

Если в параметре имеется параметр LPU\_id необходимо производить проверку на соответствие значения LPU\_id из метода и установленного в сессии. Если значения не идентичны, то метод должен генерировать ошибку.

Исключения метода:

**GET api/Refbook/Lpu**

**GET api/user/MO**

**GET api/Lpu/LpuList**

**GET api/MedSpecOms/MedSpecOmsByMO**

**GET api/MedStaffFact/MedStaffFactByMO**

**GET api/TimeTableGraf/TimeTableGrafListbyMO**

**GET api/TimeTableGraf/TimeTableGrafbyMO**

**GET api/Lpu/LpuSectionListByMO  
GET api/Lpu/LpuSectionListByBuilding**

**GET api/Lpu/LpuBuildingListByCodeAndName**

**GET api/Lpu/LpuBuildingListByMO**

**GET api/lpu/LpuSectionListByCodeAndName   
GET api/Lpu/LpuSectionListByBuilding**

**GET api/Lpu/LpuUnitById**

**GET api/Lpu/LpuSectionDopProfileList**

**GET api/Lpu/LpuSectionById**

* + 1. Получение списка МО, связанных c пользователем

**GET api/user/MO**

Список МО пользователя

|  |  |
| --- | --- |
| **Входящие параметры** | * Sess\_id (T [], О) – идентификатор сессии |
| **Ответ** | Успешный ответ – список МО   * Lpu\_id (N,O) Идентификатор МО * Org\_Name (S [300], O) - Наименование МО * Org\_Nick (S[300], 0) - Краткое наименование ЛПУ |

* + 1. Установка МО редактирования/просмотра

**PUT api/user/ReviewMO**

Установка МО просмотра. При установке МО просмотра результаты запросов будут передаваться данные по выбранной МО.

Редактирование данных, также, доступно только по МО просмотра. Если МО просмотра и МО в методах будут различны, то сервис должен генерировать ошибки.

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,O) Идентификатор МО |
| **Ответ** | Успешный ответ – ошибка с кодом 0  Пользователь не имеет права на просмотр данных по указанной МО |

* + 1. Получение параметров сессии

**GET api/user/SessionParams**

|  |  |
| --- | --- |
| **Входящие параметры** | * PHPSESSID (T, О) – идентификатор сессии |
| **Ответ** | Успешный ответ – параметры данной сессии:   * PMUser\_id (N, О) – Идентификатор пользователя; * PMUser\_Name (T, О) – Имя пользователя; * PMUser\_Login (T, О) – Логин; * MedPersonal\_id (N, О) – Идентификатор мед. работника; * Email (T, Н) – Адрес электронной почты; * Region\_id (N, О) – Идентификатор региона, к которому подключена сессия; * Lpu\_id (N, Н) – Идентификатор МО, к которой подключена сессия; * Org\_id (N, Н) – Идентификатор организации, к которой подключена сессия; * OrgType\_SysNick (T, Н) – Системное наименование типа организации; * Language (T, Н) – Язык. |

* + 1. Типы справочников, используемых в Системе

Используются следующие типы справочников:

* Федеральные: **Справочник Реестра НСИ** – это федеральные стандартизированные справочники. Опубликованы на сайте Реестра НСИ Минздрава РФ ([http://nsi.rosminzdrav.ru/index.php?show=reestr#](http://nsi.rosminzdrav.ru/index.php?show=reestr)); **Внутренний справочник ИЭМК** – это внутренние справочники ИЭМК. Опубликованы на Портале оперативного взаимодействия участников ЕГИСЗ (<http://portal.egisz.rosminzdrav.ru/materials>);
* Внутренние: **справочник** – это внутренние справочники Системы. Для получения данных по внутренним справочникам используются REST-методы.

Т а б л и ц а 1

| **Имя справочника** | **Тип справочника** | **Код справочника** | **Наименование справочника в Реестре НСИ/ИЭМК** |
| --- | --- | --- | --- |
| Медицинские организации | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.178 | Регистр медицинских организаций Российской Федерации. Версия 2. |
| Тип здания МО | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.182 | Классификатор зданий (сооружений) здравоохранения и социального обслуживания |
| Тип подразделения МО | Внутренний справочник | dbo.LpuUnitType |  |
| Дополнительные типы отделений | Внутренний справочник | dbo.LpuUnitTypeDop |  |
| Тип отделения МО | Внутренний справочник | dbo.LpuSectionType |  |
| Профиль отделения МО | Внутренний справочник | dbo.LpuSectionProfile |  |
| Вид медицинской помощи | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.103 | Классификатор медицинской помощи по условиям оказания |
| Тип отделения по возрасту | Внутренний справочник | dbo.LpuSectionAge |  |
| Тип участка | Внутренний справочник | dbo.LpuRegionType |  |
| Тип палаты | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.231 | Классификатор номерного фонда для предоставления санаторно-курортного лечения |
| Профиль коек | Справочник Реестра НСИ |  | Номенклатура коечного фонда медицинской организации |
| Уровень МЭС | Внутренний справочник | dbo.MesLevel |  |
| Тип службы | Внутренний справочник | dbo.MedServiceType |  |
| Цель проф. Осмотра | Внутренний справочник | dbo.DispProfGoalType |  |
| Типы занятости | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.204 | Классификатор видов занятости. |
| Порядок выдачи | Внутренний справочник | dbo.StickOrder |  |
| Причины нетрудоспособности | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.121 | Классификатор причин выдачи документа временной нетрудоспособности |
| Дополнительные причины нетрудоспособности | Внутренний справочник | dbo.StickCauseDopType |  |
| Типы получателей справок учащегося | Внутренний справочник | dbo.StickRecipient |  |
| Нарушения режима | Внутренний справочник | dbo.StickIrregularity |  |
| Группы инвалидности | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.702 | Группы инвалидности. |
| Исходы ЛВН | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.532 | Классификатор статусов состояний документа временной нетрудоспособности. |
| Форма рецепта | Внутренний справочник | dbo.ReceptForm |  |
| Типы выписки рецепта | Внутренний справочник | dbo.ReceptType |  |
| Актуальности рецепта | Внутренний справочник | dbo.ReceptValid |  |
| Диагноз | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.641 | Международная классификация болезней и состояний, связанных со здоровьем |
| Типы финансирования | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.76 | Классификатор уровней бюджетного финансирования льготного обеспечения граждан лекарственными средствами. |
| Типы льгот | Внутренний справочник | dbo.PrivilegeType |  |
| МНН | Внутренний справочник | dbo.DrugMnn |  |
| Аптеки | Внутренний справочник | dbo.OrgFarmacy |  |
| Препараты | Внутренний справочник | rls.DrugPrep |  |
| Единицы измерения товара | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.331 | Классификатор типов лекарственных форм. |
| Тип бирки | Внутренний справочник | dbo.TimeTableType |  |
| Руководящие должности | Внутренний справочник | dbo.OrgHeadPost |  |
| Типы документов | Внутренний справочник | dbo.XmlType |  |
| Пол | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.156 | Классификатор половой принадлежности |
| Социальный статус | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.366 | Классификатор социальных статусов населения. |
| Цель посещения | Внутренний справочник | dbo.VizitType |  |
| Вид посещения | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.83 | Классификатор видов посещений специалистов в медицинской организации. |
| Вид обращения | Внутренний справочник | dbo.TreatmentClass |  |
| Место обслуживания | Внутренний справочник | dbo.ServiceType |  |
| Вид оплаты | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.104 | Классификатор посещений по видам оплат. |
| Комплексная услуга | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.473 | Номенклатура медицинских услуг |
| Тип проффессионального осмотра | Внутренний справочник | dbo.ProfGoal |  |
| Виды диспансеризации | Внутренний справочник | dbo.DispClass |  |
| Характер заболевания | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.102 | Классификатор характера заболевания (устаревшее) |
| Категория услуг | Внутренний справочник | dbo.UslugaCategory |  |
| Тип осложнения | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.139 | Классификатор осложнений операции |
| Период, в котором произошло осложнение | Внутренний справочник | dbo.AggWhen |  |
| Место оказания услуги | Внутренний справочник | dbo.UslugaPlace |  |
| Тип операции | Внутренний справочник | dbo.OperType |  |
| Категория сложности операции | Внутренний справочник | dbo.OperDiff |  |
| Условия лечения | Внутренний справочник | dbo.TreatmentConditionsType |  |
| Тип анестезии | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.137 | Классификатор услуг анестезии по способу введения |
| Кем направлен | Внутренний справочник | dbo.PrehospDirect |  |
| Тип госпитализации | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.568 | Классификатор форм оказания медицинской помощи |
| Способ доставки | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.281 | Классификатор каналов госпитализации в лечебно-профилактическую организацию |
| Вид отравления | Внутренний справочник | dbo.PrehospToxic |  |
| Вид травмы | Внутренний справочник | dbo.PrehospTrauma |  |
| Причина отказа от госпитализации | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.194 | Классификатор причин отказов в госпитализации. |
| Форма оказания медицинской помощи | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.568 | Классификатор форм оказания медицинской помощи |
| Стадия заболевания | Внутренний справочник | dbo.DiagSetPhase |  |
| Вид тарифа | Внутренний справочник | dbo.TariffClass |  |
| МЭС | Внутренний справочник | dbo.Mes |  |
| Исход госпитализации | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.542 | Классификатор исходов госпитализации |
| Исход заболевания | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.688 | Новая версия Классификатор результатов госпитализации (исхода заболевания, причин выписки). Версия 2. |
| Причина выписки | Внутренний справочник | dbo.LeaveCause |  |
| Итог лечения | Внутренний справочник | dbo.CureResult |  |
| Тип диагноза дополнительной диспансеризации | Внутренний справочник | dbo.DopDispDiagType |  |
| Стадия заболевания (диспансеризация) | Внутренний справочник | dbo.DeseaseStage |  |
| Наследственность | Внутренний справочник | dbo.HeredityType |  |
| Тип диагноза | Внутренний справочник | dbo.DiagSetClass |  |
| Группа здоровья | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.118 | Классификатор групп состояния здоровья взрослого населения. |
| Тип полиса | Внутренний справочник | fed.PolisType |  |
| Результат диспансеризации | Внутренний справочник | fed.DispResultType |  |
| Тип диспансеризации | Внутренний справочник | fed.DispType |  |
| Метод высокотехнологичной медицинской помощи | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.446 | Классификатор методов лечения с применением высоких медицинских технологий. |
| Вид высокотехнологичной медицинской помощи | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.417 | Перечень видов высокотехнологичной медицинской помощи. |
| Профиль оказанной медицинской помощи | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.449 | Классификатор профилей (видов) специализированной медицинской помощи. |
| Вид медицинской помощи | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.655 | Классификатор медицинской помощи по видам оказания. |
| Способ оплаты медицинской помощи | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.80 | Классификатор единиц учета медицинской помощи. |
| Исход заболевания | Внутренний справочник | fed.ResultDeseaseType |  |
| Тип документа | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.498 | Классификатор документов, удостоверяющих личность гражданина Российской Федерации |
| Территория страхования | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.661 | Классификатор адресов России |
| Форма полиса | Внутренний справочник | dbo.PolisFormType |  |
| Организации выдачи | Внутренний справочник | dbo.OrgDep |  |
| Должности | Внутренний справочник | persis.Post – Ошибка явная |  |
| Страны мира | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.63 | Общероссийский классификатор стран мира |
| Запись в очередь | Внутренний справочник | dbo.RecordQueue |  |
| Общероссийский классификатор единиц измерения (ОКЕИ) | Внутренний справочник | dbo.Okei |  |
| Кем доставлен (беспризорный) | Внутренний справочник | dbo.PrehospWaifArrive |  |
| Причина помещения в МО (беспризорный) | Внутренний справочник | dbo.PrehospWaifReason |  |
| Тип примечания | Внутренний справочник | dbo.AnnotationType |  |
| Профиль вызова | Внутренний справочник | dbo.CallProfType |  |
| Тип вызова | Внутренний справочник | dbo.HomeVisitCallType |  |
| Справочник вызвавших врача | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.270 | Классификатор лиц, зафиксировавших симптомы у больного (ОНМК). |
| Симптомы | Внутренний справочник | dbo.HomeVisitSymptom |  |
| Статус вызова | Внутренний справочник | dbo.HomeVisitStatus |  |
| Вид должности | Внутренний справочник | persis.postkind |  |
| Профессиональная группа | Внутренний справочник | persis.ProfessionalGroup |  |
| Специальность | Внутренний справочник | persis.Speciality |  |
| Тип полиса | Внутренний справочник ИЭМК | 1.2.643.5.1.13.2.7.1.62 | Тип внешнего идентификатора. |
| Медикаменты | Внутренний справочник | rls.Drug |  |
| Занятость матери | Внутренний справочник | dbo.BirthEmployment |  |
| Образование матери | Внутренний справочник | dbo.BirthEducation | Классификатор образования. |
| Семейное положение | Внутренний справочник | dbo.BirthFamilyStatus |  |
| Вид мед. персонала, принимающего роды | Внутренний справочник | dbo.BirthMedPersonalType |  |
| Место принятия родов | Внутренний справочник | dbo.BirthPlace |  |
| Специалист, принимавший роды | Внутренний справочник | dbo.BirthSpecialist |  |
| Результат родов | Внутренний справочник | dbo.BirthChildResult |  |
| Тип попечителя | Внутренний справочник | dbo.DeputyKind |  |
| Тип свидетельства | Внутренний справочник | dbo.DeathSvidType |  |
| Признак доношенности | Внутренний справочник | dbo.ChildTermType |  |
| Занятость (Свидетельство о смерти) | Внутренний справочник | dbo.DeathEmployment |  |
| Образование (Свидетельство о смерти) | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.324 | Классификатор образования. |
| Тип места смерти | Внутренний справочник | dbo.DeathPlace |  |
| Семейное положение (Свидетельство о смерти) | Внутренний справочник | dbo.DeathFamilyStatus |  |
| Причина смерти | Внутренний справочник | dbo.DeathCause |  |
| Тип травмы, приведшей к смерти | Внутренний справочник | dbo.DeathTrauma |  |
| Период наступления смерти после ДТП | Внутренний справочник | dbo.DtpDeathTime |  |
| Установление причины смерти | Внутренний справочник | dbo.DeathSetType |  |
| Основание для установления причины смерти | Внутренний справочник | dbo.DeathSetCause |  |
| Период перинатальной смерти | Внутренний справочник | dbo.PntDeathPeriod |  |
| Время перинатальной смерти | Внутренний справочник | dbo.PntDeathTime |  |
| Образование (Свидетельство о перинатальной смерти) | Внутренний справочник | dbo.PntDeathEducation | Классификатор образования. |
| Семейное положение (Свидетельство о перинатальной смерти) | Внутренний справочник | dbo.PntDeathFamilyStatus |  |
| Место наступления перинатальной смерти | Внутренний справочник | dbo.PntDeathPlace |  |
| Специалист, принимавший роды (Свидетельство о перинатальной смерти) | Внутренний справочник | dbo.PntDeathGetBirth |  |
| Причина наступления перинатальной смерти | Внутренний справочник | dbo.PntDeathCause |  |
| Причина перинатальнйо смерти | Внутренний справочник | dbo.PntDeathSvidType |  |
| Специалист, установивший перинатальную смерть | Внутренний справочник | dbo.PntDeathSetType |  |
| Основание для установления причины перинатальной смерти | Внутренний справочник | dbo.PntDeathSetCause |  |
| Тип контрагента | Внутренний справочник | dbo.ContragentType |  |
| Источник финансирования | Внутренний справочник | dbo.DrugFinance |  |
| Статья расходов | Внутренний справочник | dbo.WhsDocumentCostItemType |  |
| Медицинские специальности | Внутренний справочник | dbo.MedSpecOms |  |
| Класс события | Внутренний справочник | dbo.EvnClass |  |
| Тип режима | Внутренний справочник | dbo.PrescriptionRegimeType |  |
| Тип диеты | Внутренний справочник | dbo.PrescriptionDietType |  |
| Способ применения | Внутренний справочник ИЭМК | 1.2.643.5.1.13.2.7.1.64 | Способ применения медикамента (внутренний справочник ИЭМК) |
| Вид диагноза | Внутренний справочник | dbo.DiagSetType |  |
| Видимость примечания | Внутренний справочник | dbo.AnnotationVison |  |
| Вид медицинской помощи | Внутренний справочник | persis.MedicalCareKind |  |
| Режим работы | Внутренний справочник | persis.workmode |  |
| Тип записи на прием | Внутренний справочник | dbo.RecClass |  |
| Тип записи | Внутренний справочник | dbo.RecType |  |
| Категория медицинского работника | Внутренний справочник | persis.Category |  |
| Образовательные учреждения | Внутренний справочник | persis.EducationInstitution |  |
| Тип образования | Внутренний справочник | persis.EducationType |  |
| Тип послевузовского образования | Внутренний справочник | persis.PostgraduateEducationType |  |
| Ученая степень | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.205 | Классификатор ученой степени |
| Тип больничного листа | Внутренний справочник ИЭМК | 1.2.643.5.1.13.2.7.1.37 | Тип документа о временной нетрудоспособности |
| Уровень подчиненности МО | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.636 | Справочник категорий подчиненности медицинских учреждений (ПМУ) |
| Ведомственная принадлежность | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.228 | Регистр органов государственной власти и управления |
| Форма собственности | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.200 | Общероссийский классификатор форм собственности |
| Организационно-правовая форма | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.645 | Общероссийский классификатор организационно-правовых форм |
| Тип МО для ПМУ | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.639 | Номенклатура медицинских организаций |
| Статус курорта | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.227 | Классификатор статусов курорта |
| Тип курорта | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.225 | Классификатор типов курортов по основному природному лечебному ресурсу |
| Вид экономической деятельности | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.62 | Общероссийский классификатор видов экономической деятельности |
| Вид лицензии по профилю | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.199 | Перечень работ (услуг), составляющих медицинскую деятельность, к которым предъявляются лицензионные требования |
| Тип МО по возрасту | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.78 | Типологическая номенклатура амбулаторно-поликлинических организаций |
| Местоположение учреждения | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.245 | Классификатор месторасположений санаторно-курортных организаций |
| Наименование объекта Инфраструктуры | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.238 | Номенклатура объектов инфраструктуры |
| Справочник МКБ-10 | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.641 | Международная классификация болезней и состояний, связанных со здоровьем, Десятого пересмотра. Версия 2 |
| Вид питания | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.237 | Классификатор режимов питания |
| Кратность питания | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.223 | Классификатор кратности питания |
| Форма питания | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.236 | Классификатор способов подачи питания |
| Природные лечебные факторы | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.239 | Классификатор природных лечебных ресурсов |
| Тип природного лечебного фактора | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.233 | Классификатор природных лечебных ресурсов по месту добычи |
| Документ на природный лечебный фактор | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.240 | Перечень документов, подтверждающих наличие природных лечебных факторов |
| Место использования природных лечебных факторов | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.241 | Перечень мест использования природных лечебных факторов |
| Классификатор зданий по технологиям строительства | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.230 | Классификатор зданий (сооружений) по технологии строительства |
| Назначение здания | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.224 | Классификатор зданий (сооружений) по назначению |
| Отношение к собственности | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.198 | Классификатор форм пользования имуществом |
| Тип постройки | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.234 | Классификатор зданий (сооружений) по архитектурному проекту |
| Несущие конструкции | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.647 | Классификатор конструктивных систем зданий по материалу несущих конструкций |
| Тип перекрытий | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.648 | Классификатор конструкций систем зданий по виду перекрытий |
| Здание. Конструкции. Текущее состояние здания | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.222 | Классификатор технического состояния здания |
| Горячее водоснабжение | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.100 | Классификатор систем горячего водоснабжения |
| Отопление | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.97 | Классификатор отопительных систем |
| Канализация | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.96 | Классификатор систем канализации |
| Комфортность палаты | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.244 | Классификатор объектов дополнительного оснащения номеров санаторно-курортной организации |
| Тип операции над койкой | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.232 | Классификатор типов операций над койкой (ПМУ) |
| Тип подразделения ФРМП | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.220 | Перечень отделений (кабинетов) и коек медицинской организации. |
| Должности ФРМП | Внутренний справочник | persis.FRMPSertificateSpeciality |  |
| Тип информационной системы | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.650 | Перечень используемых информационных систем |
| СМО | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.635 | Реестр страховых медицинских организаций (ФОМС). |
| Результат лечения | Внутренний справочник | dbo.ResultClass |  |
| Исход заболевания | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.688 | Новая версия Классификатор результатов госпитализации (исхода заболевания, причин выписки). Версия 2. |
| Шкала Рэнкина | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.306 | Классификатор степени нарушения функций и жизнедеятельности по шкале Рэнкин |
| Военкомат | Внутренний справочник | dbo.OrgMilitary |  |
| Территории | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.661 | Классификатор адресов России |
| Улицы | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.663 | Классификатор адресов России. Улицы |
| Тип назначения | Внутренний справочник | dbo.PrescriptionType |  |
| Статус назначения | Внутренний справочник | dbo.PrescriptionStatusType |  |
| Тип исполнения | Внутренний справочник | dbo.PerformanceType |  |
| Комплексные МНН | Внутренний справочник | dbo.DrugComplexMnn |  |
| Тип продолжительности приема | Внутренний справочник | dbo.DurationType |  |
| ОКОГУ | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.228 | Регистр органов государственной власти и управления |
| Справочник организаций | Внутренний справочник | [dbo.Org](http://fixuploader.swn.local/dbspr/index/667878192) |  |
| Уровень учреждения в иерархии сети | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.235 | Классификатор уровеней учреждений в иерархии сети |
| Наименование операции с лицензией | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.242 | Классификатор статусов состояний лицензии |
| Класс технологии | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.652 | Классификатор медицинских технологий в зависимости от степени потенциального риска применения в медицинских целях |
| Медицинские услуги | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.653 | Требования к организации и выполнению работ (услуг) в Здравоохранении |
| Период функционирования учреждения | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.243 | Классификатор периодов функционирования санаторно-курортных организаций |
| Тип расчетного счета | Внутренний справочник | [dbo.OrgRSchetType](http://fixuploader.swn.local/dbspr/index/7839986) |  |
| Организации типа Банк | Внутренний справочник | [dbo.OrgBank](http://fixuploader.swn.local/dbspr/index/1847729685) |  |
| Справочник валют | Внутренний справочник | [dbo.Okv](http://fixuploader.swn.local/dbspr/index/103840328) |  |
| Тип населенного пункта | Внутренний справочник | [dbo.KLAreaType](http://fixuploader.swn.local/dbspr/index/519724954) |  |
| ОКАТО | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.608 | Общероссийский классификатор объектов административно-территориального деления |
| Виды топлива отопления | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.649 | Классификатор видов топлива |
| Каналы связи | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.92 | Классификатор типов выделенных каналов связи |
| Приобретено в рамках | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.671 | Каналы финансирования приобретения медицинских изделий |
| Класс потенциального риска применения | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.74 | Классификация медицинских изделий в зависимости от потенциального риска применения |
| Функциональное назначение | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.672 | Классификатор функционального назначения медицинских изделий |
| Типы медицинских изделий | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.684 | Типы медицинских изделий (Медтехника) |
| Область медицинского применения | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.67 | Классификатор областей медицинского применения медицинских изделий |
| Сфера применения | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.679 | Сфера применения медицинских изделий |
| Вид медицинского изделия | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.61 | Классификатор медицинских изделий (изделий медицинского назначения и медицинской техники) |
| 30й ФЗ | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.66 | Перечень аппаратов и оборудования отделений (кабинетов) медицинской организации |
| ТН ВЭД | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.678 | ГРУППА 90 ИНСТРУМЕНТЫ И АППАРАТЫ ОПТИЧЕСКИЕ, ФОТОГРАФИЧЕСКИЕ, КИНЕМАТОГРАФИЧЕСКИЕ, ИЗМЕРИТЕЛЬНЫЕ, КОНТРОЛЬНЫЕ, ПРЕЦИЗИОННЫЕ, МЕДИЦИНСКИЕ ИЛИ ХИРУРГИЧЕСКИЕ; ИХ ЧАСТИ И ПРИНАДЛЕЖНОСТИ |
| GMDN | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.673 | Всемирная номенклатура медицинских изделий (GMDN) |
| Классификатор МТ по 97 приказу | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.683 | Перечень изделий медицинской техники в соответствии с формой заявки на приобретение медицинской техники. Версия 2 |
| ОКОФ | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.680 | Общероссийский классификатор основных фондов |
| ОКП | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.675 | Класс 94 МЕДИЦИНСКАЯ ТЕХНИКА |
| ОКПД | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.677 | Класс 33 Изделия медицинские |
| Тип поставки | Внутренний справочник | passport.DeliveryType |  |
| Единица измерения | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.674 | Перечень единиц измерения медицинских изделий (Медтехника) |
| Причина простоя | Внутренний справочник | passport.DowntimeCause |  |
| Лекарственная форма | Справочник Реестра НСИ | 1.2.643.5.1.13.2.1.1.331 | Классификатор типов лекарственных форм |
| Тип занятия должности | Внутренний справочник РИШ | persis.PostOccupationType |  |
| Тип приказа о приеме | Внутренний справочник РИШ | persis.ArriveRecordType |  |
| Справочник должностей ФРМП | Внутренний справочник РИШ | persis.frmppost |  |
| Отношение к воинской службе | Внутренний справочник РИШ | persis.MilitaryRelation |  |
| Тип подразделения ФРМП | Внутренний справочник РИШ | persis.FRMPSubdivision |  |
| Тип приказа об увольнении локальный | Внутренний справочник РИШ | persis.LeaveRecordType |  |
| Тип заболевания | Внутренний справочник РИШ | dbo.MorbusType |  |
| Справочник типов лучевой терапии | Внутренний справочник РИШ | dbo.OnkoRadiotherapy |  |
| Характер лечения | Внутренний справочник РИШ | dbo.OnkoTreatType |  |
| Группы диспансерного наблюдения | Внутренний справочник РИШ | dbo.PersonDispGroup |  |
| Тип регистра | Внутренний справочник РИШ | dbo.PersonRegisterType |  |
| Результат консультации | Внутренний справочник РИШ | dbo.TubAdviceResultType |  |
| Причина прерывания химиотерапии | Внутренний справочник РИШ | dbo.TubBreakChemType |  |
| Тип исхода курса химиотерапии | Внутренний справочник РИШ | dbo.TubResultChemType |  |
| Причина смерти | Внутренний справочник РИШ | dbo.TubResultDeathType |  |
| Фактор риска: Туберкулез | Внутренний справочник РИШ | dbo.TubRiskFactorType |  |
| Фазы химиотерапии | Внутренний справочник РИШ | dbo.TubStageChemType |  |
| Стандартные режимы химиотерапии | Внутренний справочник РИШ | dbo.TubStandartConditChemType |  |
| Типы групп наблюдаемых в тубучреждениях | Внутренний справочник РИШ | dbo.TubSurveyGroupType |  |
| Тип места проведения | Внутренний справочник РИШ | dbo.TubVenueType |  |
| Результат аутопсии применительно к данной опухоли | Внутренний справочник РИШ | dbo.TumorAutopsyResultType |  |
| Аутопсия | Внутренний справочник РИШ | dbo.AutopsyPerformType |  |
| Справочник диагностических показателей | Внутренний справочник РИШ | dbo.DiagAttribDict |  |
| Справочник типов диагностического показателя | Внутренний справочник РИШ | dbo.DiagAttribType |  |
| Справочник результатов диагностики | Внутренний справочник РИШ | dbo.DiagResult |  |
| Тестирование на лекарственную устойчивость | Внутренний справочник РИШ | dbo.DrugResistenceTest |  |
| Справочник линий лекарственной терапии | Внутренний справочник РИШ | dbo.DrugTherapyLineType |  |
| Справочник циклов лекарственной терапии | Внутренний справочник РИШ | dbo.DrugTherapyLoopType |  |
| Тип контингента | Внутренний справочник РИШ | dbo.HIVContingentType |  |
| Причина снятия с диспансерного наблюдения | Внутренний справочник РИШ | dbo.HIVDispOutCauseType |  |
| Тип вируса | Внутренний справочник РИШ | dbo.HIVInfectType |  |
| Стадии ВИЧ-инфекции | Внутренний справочник РИШ | dbo.HIVPregInfectStudyType |  |
| предполагаемый путь инфицирования | Внутренний справочник РИШ | dbo.HIVPregPathTransType |  |
| Результаты лабораторных исследований | Внутренний справочник РИШ | dbo.LabAssessmentResult |  |
| Молекулярно-генетические методы:Тест на лекарственную устойчивость | Внутренний справочник РИШ | dbo.MorbusTubMolecularType |  |
| Сочетание видов лечения | Внутренний справочник РИШ | dbo.OnkoCombiTreatType |  |
| Морфологическая классификация новообразований (МКБ-0) | Внутренний справочник РИШ | dbo.OnkoDiag |  |
| Метод подтверждения диагноза по онкологии | Внутренний справочник РИШ | dbo.OnkoDiagConfType |  |
| Онкология: инвалидность по общему (онкологическому) заболеванию | Внутренний справочник РИШ | dbo.OnkoInvalidType |  |
| Позднее осложнение специального лечения | Внутренний справочник РИШ | dbo.OnkoLateComplTreatType |  |
| Причины поздней диагностики | Внутренний справочник РИШ | dbo.OnkoLateDiagCause |  |
| Сторона поражения | Внутренний справочник РИШ | dbo.OnkoLesionSide |  |
| Вид планирования лучевого лечения | Внутренний справочник РИШ | dbo.OnkoPlanType |  |
| Должность врача установившего заболевание | Внутренний справочник РИШ | dbo.OnkoPostType |  |
| Состояние на конец отчетного года | Внутренний справочник РИШ | dbo.OnkoStatusYearEndType |  |
| Справочник типов хирургического лечения | Внутренний справочник РИШ | dbo.OnkoSurgicalType |  |
| Характер хирургического лечения | Внутренний справочник РИШ | dbo.OnkoSurgTreatType |  |
| Справочник поводов обращений при онкологическом заболевании | Внутренний справочник РИШ | dbo.OnkoTreatment |  |
| Состояние опухолевого процесса (мониторинг опухоли) | Внутренний справочник РИШ | dbo.OnkoTumorStatusType |  |
| Преимущественная направленность лучевой терапии | Внутренний справочник РИШ | dbo.OnkoUslugaBeamFocusType |  |
| Способ облучения при проведении лучевой терапии | Внутренний справочник РИШ | dbo.OnkoUslugaBeamIrradiationType |  |
| Вид лучевой терапии | Внутренний справочник РИШ | dbo.OnkoUslugaBeamKindType |  |
| Метод лучевой терапии | Внутренний справочник РИШ | dbo.OnkoUslugaBeamMethodType |  |
| Радиомодификаторы | Внутренний справочник РИШ | dbo.OnkoUslugaBeamRadioModifType |  |
| Единица измерения | Внутренний справочник РИШ | dbo.OnkoUslugaBeamUnitType |  |
| Преимущественная направленность химиотерапии | Внутренний справочник РИШ | dbo.OnkoUslugaChemFocusType |  |
| Вид проведенного химиотерапевтического лечения | Внутренний справочник РИШ | dbo.OnkoUslugaChemKindType |  |
| Этапы лечения по химиотерапии | Внутренний справочник РИШ | dbo.OnkoUslugaChemStageType |  |
| Преимущественная направленность гормоноиммунотерапии | Внутренний справочник РИШ | dbo.OnkoUslugaGormunFocusType |  |
| Категория населения | Внутренний справочник РИШ | dbo.PersonCategoryType |  |
| Причины смерти | Внутренний справочник РИШ | dbo.PersonDeathCause |  |
| Декретированные группы | Внутренний справочник РИШ | dbo.PersonDecreedGroup |  |
| Жилищные условия | Внутренний справочник РИШ | dbo.PersonLivingFacilies |  |
| Справочник причин невключения в регистр | Внутренний справочник РИШ | dbo.PersonRegisterFailIncludeCause |  |
| Справочник причин исключения из регистра | Внутренний справочник РИШ | dbo.PersonRegisterOutCause |  |
| Статус пациента | Внутренний справочник РИШ | dbo.PersonResidenceType |  |
| Бактериовыделение | Внутренний справочник РИШ | dbo.TubBacterialExcretion |  |
| Классификатор клинических форм больных туберкулезом | Внутренний справочник РИШ | 1.2.643.5.1.13.13.99.2.65 | Классификатор клинических форм больных туберкулезом |
| Обстоятельства, при которых выявлено заболевание (пути выявления) | Внутренний справочник РИШ | dbo.TubDetectionFactType |  |
| Метод выявления | Внутренний справочник РИШ | dbo.TubDetectionMethodType |  |
| Место выявления | Внутренний справочник РИШ | dbo.TubDetectionPlaceType |  |
| Заболевание по форме №8 | Внутренний справочник РИШ | dbo.TubDiagForm8 |  |
| Диагностический материал | Внутренний справочник РИШ | dbo.TubDiagnosticMaterialType |  |
| Диагноз в извещении | Внутренний справочник РИШ | dbo.TubDiagNotify |  |
| Сопутствующее заболевание | Внутренний справочник РИШ | dbo.TubDiagSop |  |
| Противотуберкулезные препараты | Внутренний справочник РИШ | dbo.TubDrug |  |
| Сроки предыдущего ФГ обследования | Внутренний справочник РИШ | dbo.TubFluorSurveyPeriodType |  |
| Результат гистологии | Внутренний справочник РИШ | dbo.TubHistolResultType |  |
| Классификатор локализации туберкулеза | Внутренний справочник РИШ | 1.2.643.5.1.13.13.99.2.83 | Классификатор локализации туберкулеза |
| Метод подтверждения бактериовыделения | Внутренний справочник РИШ | dbo.TubMethodConfirmBactType |  |
| Результаты микроскопии | Внутренний справочник РИШ | dbo.TubMicrosResultType |  |
| Тип учета в наркологическом диспансере | Внутренний справочник РИШ | dbo.TubRegCrazyType |  |
| Результат посева | Внутренний справочник РИШ | dbo.TubSeedResultType |  |
| Группы больных | Внутренний справочник РИШ | dbo.TubSickGroupType |  |
| Цель исследования | Внутренний справочник РИШ | dbo.TubTargetStudyType |  |
| Результат рентгена | Внутренний справочник РИШ | dbo.TubXrayResultType |  |
| Обстоятельства выявления опухоли | Внутренний справочник РИШ | dbo.TumorCircumIdentType |  |
| Первично-множественная опухоль | Внутренний справочник РИШ | dbo.TumorPrimaryMultipleType |  |
| Стадия опухолевого процесса | Внутренний справочник РИШ | dbo.TumorStage |  |
| Способ введения вакцины | Внутренний справочник РИШ | Vac.S\_VaccineWay |  |
| Место введения вакцины | Внутренний справочник РИШ | Vac. S\_VaccinePlace |  |
| Тип реакции манту | Внутренний справочник РИШ | vac.S\_MantuReactionType |  |
| Тип туберкулиновой пробы | Внутренний справочник РИШ | vac.S\_TubDiagnosisType |  |
| Тип реакции диаскин тест | Внутренний справочник РИШ | vac.S\_DiaskinTestReactionType |  |
| Коды отделений | Внутренний справочник РИШ | dbo.LpuSectionCode |  |

* + 1. Получение списка справочников

**GET api/RefbookList**

|  |  |
| --- | --- |
| **Входящие параметры** | Без входных параметров |
| **Ответ** | Выводится список справочников со следующими параметрами:   * Refbook\_Code (T, О) – Код справочника; * Refbook\_TableName (T, О) – Наименование таблицы справочника; * Refbook\_Name (T, О) – Наименование справочника; * RefbookType\_id (N, О) – Тип справочника. |

* + 1. Получение элементов справочника

**GET api/Refbook**

|  |  |
| --- | --- |
| **Входящие параметры** | * Refbook\_Code (T, Н) – Код справочника; * Refbook\_TableName (T, Н)– Наименование таблицы справочника; * id (N, Н) – Ид элемента справочника; * Code (T, Н) – Код элемента справочника * Name (T, Н) – Наименование элемента справочника |
| **Ответ** | 1. Проводится проверка: Хотя бы один из параметров Refbook\_Code, Refbook\_TableName должен быть задан. Если условие не выполняется, то ошибка. 2. Иначе производится поиск справочника по Коду справочника (Refbook\_Code) и/или Наименованию таблицы справочника (Refbook\_TableName). 3. При поиске по коду справочника если найдено более одной записи, то в ответ возвращается ошибка: «По указанному коду <Код справочника> найдено несколько справочников. Уточните запрос данными о наименовании таблицы (Refbook\_TableName)». 4. Если найдена одна запись, то выводится список элементов справочника, отфильтрованный по параметрам id, Code, Name (если указаны):  * id – Идентификатор элемента справочника; * Name – Наименование элемента справочника; * Code – Код элемента справочника * Дата начала * Дата окончания |

* + 1. Получение элементов справочника с учетом таблиц стыковки

**GET api/RefbookMap**

|  |  |
| --- | --- |
| **Входящие параметры** | * Refbook\_Code (T, О) – Код справочника; * Refbook\_MapName (T, Н) – Наименование таблицы стыковки; * Column\_Name (N, Н) – Наименование поля таблицы стыковки, по которому будет производиться фильтрация записей. Может принимать следующие значения:   + «id» - тогда фильтрация должна происходить по Ид справочника ЕЦП.   + «Code» - тогда фильтрация должна происходить по Коду справочника ЕЦП.   + «Fed\_id» - тогда фильтрация должна происходить по Ид федерального справочника.   + «Fed\_code» - тогда фильтрация должна происходить по Коду федерального справочника. * Column\_Value (T, Н) – Значение поля таблицы стыковки для фильтрации записей. |
| **Ответ** | Алгоритм следующий:   1. В таблице справочников dbo.RISHRefbook по Коду справочника (Refbook\_Code) и наименованию таблицы стыковки (Refbook\_MapName, если параметр задан) определяется таблица стыковки. 2. Если таблица стыковки не найдена, то в ответе возвращается «0». 3. Иначе в таблице стыковки производится поиск записей, удовлетворяющих условию: Column\_Name = Column\_Value.   Если какой-то из двух параметров (Column\_Name, Column\_Value) не задан, то фильтрация не производится.  В ответе выводится список элементов таблицы стыковки со следующими полями:   * id – Идентификатор элемента справочника РИШ; * Code – Код элемента справочника РИШ; * Fed\_id – Идентификатор федерального справочника; * Fed\_Code – Код федерального справочника. |

* + 1. Получение элементов справочника СМО (расширенный)

**GET api/Refbook/OrgSMO**

|  |  |
| --- | --- |
| **Входящие параметры** | * OrgSMO\_id (N, Н) – Идентификатор СМО в справочнике РИШ; * OrgSmo\_Name (T, Н) – Наименование СМО; * Orgsmo\_f002smocod (T, Н) – Код СМО в федеральном справочнике; * OrgSMO\_Fedid (N, Н) – Идентификатор СМО в федеральном справочнике; * KLRgn\_id (N, Н) – Идентификатор региона. |
| **Ответ** | Выводится список элементов справочника dbo.OrgSMO, отфильтрованный по параметрам (если указаны):   * OrgSMO\_id (N, О) – Идентификатор СМО в справочнике РИШ; * OrgSmo\_Name (T, Н) – Наименование СМО; * Orgsmo\_f002smocod (T, Н) – Код СМО в федеральном справочнике; * OrgSMO\_Fedid (N, Н) – Идентификатор СМО в федеральном справочнике; * KLRgn\_id (N, О) – Идентификатор региона. |

* + 1. Получение элементов справочника МО (расширенный)

**GET api/Refbook/Lpu**

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N, Н) – Идентификатор МО в справочнике РИШ; * Lpu\_Name (T, Н) – Наименование МО; * Lpu\_f003mcod (T, Н) – Код МО в федеральном справочнике; * LPU\_OID (T, Н) – OID МО в федеральном справочнике (значение поля PassportToken\_tid справочника fed.PassportToken). |
| **Ответ** | Выводится список элементов справочника dbo. Lpu, отфильтрованный по параметрам (если указаны):   * Lpu\_id (N, Н) – Идентификатор МО в справочнике РИШ; * Lpu\_Name (T, Н) – Наименование МО; * Lpu\_f003mcod (T, Н) – Код МО в справочнике ФОМС; * LPU\_OID (T, Н) – OID МО в федеральном справочнике (значение поля PassportToken\_tid справочника fed.PassportToken). |

* + 1. Получение элементов справочника территорий КЛАДР (расширенный)

**GET api/Refbook/KLArea**

|  |  |
| --- | --- |
| **Входящие параметры** | * KLAdr\_Code (N, О) – Код КЛАДР |
| **Ответ** | По коду КЛАДР определяется список элементов справочника dbo.KLArea с указанием параметров:   * KLCountry\_id (N, О) – Идентификатор страны; * KLArea\_id (N, О) – Идентификатор территории; * KLRgn\_id (N, Н) – Идентификатор региона. Определяется по алгоритму:   + Если KLAreaLevel\_id = 1. Регион, то KLRgn\_id = KLArea\_id;   + Иначе через KLArea\_pid нужно выйти на уровень, где KLAreaLevel\_id = 1. Регион. KLArea\_id найденной записи = KLRgn\_id. * KLSubRgn\_id (N, Н) – Идентификатор района. Определяется по алгоритму:   + Если KLAreaLevel\_id = 2. Район, то KLSubRgn\_id = KLArea\_id;   + Иначе через KLArea\_pid нужно выйти на уровень, где KLAreaLevel\_id = 2. Район. KLArea\_id найденной записи = KLSubRgn\_id. * KLCity\_id (N, Н) – Идентификатор города. Определяется по алгоритму:   + Если KLAreaLevel\_id = 3. Город, то KLCity\_id = KLArea\_id;   + Иначе через KLArea\_pid нужно выйти на уровень, где KLAreaLevel\_id = 3. Город. KLArea\_id найденной записи = KLCity\_id. * KLTown\_id (N, Н) – Идентификатор населенного пункта. Определяется по алгоритму:   + Если KLAreaLevel\_id = 4. Населенный пункт, то KLTown\_id = KLArea\_id;   + Иначе через KLArea\_pid нужно выйти на уровень, где KLAreaLevel\_id = 4. Населенный пункт. KLArea\_id найденной записи = KLTown\_id. |

* + 1. Получение элементов справочника улиц КЛАДР (расширенный)

**GET api/Refbook/KLStreet**

|  |  |
| --- | --- |
| **Входящие параметры** | * KLStreet\_Name (T, О) – Наименование улицы. * KLArea\_id (N, Н) – Идентификатор территории. |
| **Ответ** | По наименованию улицы и идентификатору территории определяется список элементов справочника dbo.KLStreet по следующему алгоритму:   1. Поиск улицы производится по наименованию улицы (без указания типа улицы).   where KLStreet\_Name = @KLStreet\_Nick   1. Если НЕ найдено ни одной записи, то поиск улицы произвоится по наименованию улицы с учетом типа улицы.   where KLStreet\_Name + ' ' + KLSocr\_Nick = @KLStreet\_Name   1. Если НЕ найдено ни одной записи, то поиск улицы производится по наименованию улицы с учетом типа улицы и знака «.».   where KLStreet\_Name + ' ' + KLSocr\_Nick + '.' = @KLStreet\_Name  Успешный ответ – список улиц с параметрами:   * KLStreet\_id (N, О) – Идентификатор улицы; * KLStreet\_Name (T, О) – Наименование улицы.   Пример: улица «Абрикосовая» должна быть найдена в случае, если во входном параметре KLStreet\_Name передано любое из следующих значений:  «Абрикосовая», «Абрикосовая УЛ», «Абрикосаовая УЛ.», «Абрикосовая ул», «Абрикосовая ул.» и т.д. |

* + 1. Получение списка районов

**GET api/Refbook/KLSubRgn**

|  |  |
| --- | --- |
| **Входящие параметры** | * KLAdr\_Code (N, О) – Код КЛАДР |
| **Ответ** | Если переданный код КЛАДР KLAdr\_Code соответствует Региону, (KLAreaLevel\_id = 1) то   1. По переданному KLAdr\_Code определяется KLArea\_id 2. Определяется список идентификаторов районов (KLAreaLevel\_id = 2) у которых KLArea\_pid = KLArea\_id. 3. Если не найдено ни одного района, то метод возвращает ошибку   иначе (KLAreaLevel\_id <> 1) метод возвращает ошибку  **Успешный ответ метода**  список Территорий (районов) с параметрами   * KLArea\_id (N, О) – Идентификатор территорий * KLArea\_Name (T, О) - наименование территории (региона) * KLSocr\_id (N, O)- идентификатор типа территории (справочник KLSocr) * KLSocr\_Name (T, O)- наименование типа территории (справочник KLSocr) * KLSocr\_Nick (T, O) - краткое наименование типа территории (справочник KLSocr) |
| **Пример** |  |

* + 1. Получение элементов справочника территорий страхования (расширенный)

**GET api/RefbookOmsSprTerr**

|  |  |
| --- | --- |
| **Входящие параметры** | * KLAdr\_Code (N, О) – Код КЛАДР территории страхования |
| **Ответ** | Алгоритм:   1. По коду КЛАДР в справочнике dbo.KLArea определяется Идентификатор территории (KLArea\_id) и уровень территории (KLAreaLevel\_id); 2. Если KLAreaLevel\_id = 1. Регион, то в справочнике территорий страхования dbo.OmsSprTerr определяется OmsSprTerr\_id, где KLRgn\_id = KLArea\_id (если значений несколько, то берется первое); 3. Если KLAreaLevel\_id = 2. Район, то в справочнике территорий страхования dbo.OmsSprTerr определяется OmsSprTerr\_id, где KLSubRgn\_id = KLArea\_id (если значений несколько, то берется первое); 4. Если KLAreaLevel\_id = 3. Город, то в справочнике территорий страхования dbo.OmsSprTerr определяется OmsSprTerr\_id, где KLCity\_id = KLArea\_id (если значений несколько, то берется первое); 5. Если KLAreaLevel\_id = 4. Населенный пункт, то в справочнике территорий страхования dbo.OmsSprTerr определяется OmsSprTerr\_id, где KLRgn\_id = KLArea\_id (если значений несколько, то берется первое).   В ответе возвращается:   * OmsSprTerr\_id (N, О) – Идентификатор территории страхования; * OmsSprTerr\_Name (T, Н) – Наименование территории страхования |

* + 1. Получение элементов справочника услуг

**GET api/RefbookUslugaComplex**

|  |  |
| --- | --- |
| **Входящие параметры** | * UslugaComplex\_id (N, Н) – Идентификатор услуги; * UslugaComplex\_pid (N, Н) – Идентификатор услуги-родителя; * UslugaComplexLevel\_id (N, Н) – Уровень услуги; * Lpu\_id (N, Н) – МО (значение справочника dbo. Lpu); * LpuSection\_id (N, Н) – Отделение МО (значение сущности dbo. LpuSection); * UslugaComplex\_Code (T, Н) – Код услуги; * UslugaComplex\_Name (T, Н) – Наименование услуги; * UslugaCategory\_id (N, Н) – Категория услуги (значение справочника dbo.UslugaCategory) |
| **Ответ** | Алгоритм:  Если входной параметр метода Lpu\_id <> Lpu\_id из параметров сессии (метод GET api/user/SessionParams), то ошибка.  Если не передан ни один параметр, то ошибка.  Если не найдено ни одной записи, то ответ «0».  В ответе выводятся все записи из dbo.UslugaComplex, у которых:   1. Если входной параметр метода Lpu\_id заполнен, то фильтруются записи, у которых Lpu\_id = входной параметр метода Lpu\_id; 2. Если входной параметр метода Lpu\_id НЕ заполнен, то фильтруются записи, у которых Lpu\_id НЕ заполнено ИЛИ Lpu\_id = Lpu\_id из параметров сессии.   В успешном ответе возвращается:   * UslugaComplex\_id (N, О) – Идентификатор услуги; * UslugaComplex\_pid (N, Н) – Идентификатор услуги-родителя; * UslugaComplexLevel\_id (N, Н) – Уровень услуги; * Lpu\_id (N, Н) – МО (значение справочника dbo. Lpu); * LpuSection\_id (N, Н) – Отделение МО (значение сущности dbo. LpuSection); * UslugaComplex\_Code (T, О) – Код услуги; * UslugaComplex\_Name (T, О) – Наименование услуги; * UslugaCategory\_id (N, Н) – Категория услуги (значение справочника dbo.UslugaCategory) |

* + 1. Получение элементов справочника единиц измерения

**GET api/RefbookGoodsUnit**

|  |  |
| --- | --- |
| **Входящие параметры** | * GoodsUnit\_id (N, Н) – Идентификатор единицы измерения; * GoodsUnit\_Name (Т, Н) – Наименование единицы измерения; * GoodsUnit\_Nick (T, Н) – Краткое наименование единицы измерения. |
| **Ответ** | Если не передан ни один параметр, то ошибка.  Если не найдено ни одной записи, то ответ «0».  Производится поиск записей в таблице dbo.GoodsUnit.  Если хотя бы одна запись найдена, то успешный ответ – список единиц измерения с параметрами:   * GoodsUnit\_id (N, О) – Идентификатор единицы измерения; * GoodsUnit\_Name (Т, О) – Наименование единицы измерения; * GoodsUnit\_Nick (T, О) – Краткое наименование единицы измерения. |

* + 1. Получение элементов справочника типов льгот

**GET api/RefbookPrivilegeType**

|  |  |
| --- | --- |
| **Входящие параметры** | * PrivilegeType\_id (N, Н) – Идентификатор типа льготы; * PrivilegeType\_Code (N, Н) – Код типа льготы; * PrivilegeType\_Name (T, Н) – Наименование типа льготы; * PrivilegeType\_Descr (T, Н) – Описание типа льготы; * ReceptDiscount\_id (N, Н) – Идентификатор скидки; * ReceptFinance\_id (N, Н) – Идентификатор типа финансирования. |
| **Ответ** | Если не передан ни один параметр, то ошибка.  Если не найдено ни одной записи, то ответ «0».  Производится поиск записей в таблице dbo. PrivilegeType.  Если хотя бы одна запись найдена, то успешный ответ – список единиц измерения с параметрами:   * PrivilegeType\_id (N, О) – Идентификатор типа льготы; * PrivilegeType\_Code (N, О) – Код типа льготы; * PrivilegeType\_Name (T, О) – Наименование типа льготы; * PrivilegeType\_Descr (T, Н) – Описание типа льготы; * ReceptDiscount\_id (N, О) – Идентификатор скидки; * ReceptFinance\_id (N, О) – Идентификатор типа финансирования. |

* + 1. Получение медикамента МНН

**GET api/RefbookDrugComplexMnn**

|  |  |
| --- | --- |
| **Входящие параметры** | * CLSDRUGFORMS\_ID (N, Н) – Идентификатор лекарственной формы Внутреннего справочника РИШ; * FEDDRUGFORMS\_ID (N, Н) – Идентификатор лекарственной формы Справочника Реестра НСИ; * CLSATC\_Code (T, Н) – Код действующего вещества АТХ; * Drug\_Dose (T, Н) – Доза медикамента. |
| **Ответ** | Если не передан ни один параметр, то ошибка.  Если не найдено ни одной записи, то ответ «0».  Алгоритм:   1. Если задан Ид лекарственной формы Справочника Реестра НСИ (FEDDRUGFORMS\_ID), но НЕ задан Ид лекарственной формы Внутреннего справочника РИШ (CLSDRUGFORMS\_ID), то производится поиск Ид Внутреннего справочника РИШ с помощью таблицы стыковки справочников; 2. Производится поиск медикаментов МНН второго уровня (и параметр DrugComplexMnn\_pid – ссылка на МНН первого уровня) с помощью запроса:   select top 1 mnn.DrugComplexMnn\_pid  from rls.DRUG dr  inner join rls.NOMEN no on no.NOMEN\_ID=dr.Drug\_id  inner join rls.PREP pr on pr.Prep\_id=no.PREPID  inner join rls.PREP\_ATC atc on atc.PREPID=pr.Prep\_id  inner join rls.CLSATC satc on satc.CLSATC\_ID=atc.uniqid  inner join rls.drugcomplexmnn mnn  on dr.DrugComplexMnn\_id=mnn.DrugComplexMnn\_id  where satc.NAME like '%@CLSATC\_Code%' -- Условие применяется, если параметр задан  and mnn.CLSDRUGFORMS\_ID=@FEDDRUGFORMS\_ID -- Условие применяется, если параметр задан  and Drug\_Dose = @Drug\_Dose -- Условие применяется, если параметр задан  ПРИМЕР:  select top 1 mnn.DrugComplexMnn\_pid DrugComplexMnn\_id, mnn.CLSDRUGFORMS\_ID,  from rls.DRUG dr  inner join rls.NOMEN no on no.NOMEN\_ID=dr.Drug\_id  inner join rls.PREP pr on pr.Prep\_id=no.PREPID  inner join rls.PREP\_ATC atc on atc.PREPID=pr.Prep\_id  inner join rls.CLSATC satc on satc.CLSATC\_ID=atc.uniqid  inner join rls.drugcomplexmnn mnn  on dr.DrugComplexMnn\_id=mnn.DrugComplexMnn\_id  where satc.NAME like '%J01CA04%'  and mnn.CLSDRUGFORMS\_ID=210  and Drug\_Dose = '500 мг'   1. Определяются параметры медикамента МНН первого уровня с помощью запроса:   select DrugComplexMnn\_id, DrugComplexMnn\_RusName, CLSDRUGFORMS\_ID  from rls.drugcomplexmnn  where DrugComplexMnn\_id=@DrugComplexMnn\_pid (результат предыдущего запроса)  Успешный ответ:   * DrugComplexMnn\_id (N, О) – Идентификатор медикамента МНН; * DrugComplexMnn\_RusName (T, О) – Наименование медикамента МНН; * CLSDRUGFORMS\_ID (N, Н) – Идентификатор лекарственной формы |

* + 1. Получение элементов справочника по столбцу

**GET api/RefbookbyColumn**

|  |  |
| --- | --- |
| **Входящие параметры** | * Refbook\_Code (T, О) – Код справочника; * Column\_Name (N, Н) – Наименование поля таблицы стыковки, по которому будет производиться фильтрация записей; * Column\_Value (T, Н) – Значение поля таблицы стыковки для фильтрации записей. |
| **Ответ** | Алгоритм следующий:  В справочнике производится поиск записей, удовлетворяющих условию: Column\_Name = Column\_Value.  Если какой-то из двух параметров (Column\_Name, Column\_Value) не задан, то фильтрация не производится.  В ответе выводится список элементов таблицы стыковки со следующими полями:   * id – Идентификатор элемента справочника; * Code – Код элемента справочника; * Name – Имя элемента справочника. |

* + 1. Получение элементов справочника по столбцу (расширенный)

**GET api/RefbookbyColumnExt**

|  |  |
| --- | --- |
| **Входящие параметры** | * Refbook\_Code (T, H) – Код справочника; * Refbook\_TableName(T, H)- наименование таблицы * Column\_Callback (T,O) – наименование поля таблицы, значения которого необходимо вернуть. * Column\_Name (T, Н) – Наименование поля таблицы , по которому будет производиться фильтрация записей; * Column\_Value (T, Н) – Значение поля таблицы для фильтрации записей. |
| **Ответ** | Алгоритм следующий:  Если поле Column\_ Callback отсутствует в таблице Refbook\_Code, то ошибка.  В таблице производится поиск записей по полю Column\_Name с заданным значением Column\_Value.  Если какой-то из двух параметров (Column\_Name, Column\_Value) не задан, то фильтрация не производится.  В ответе выводится список элементов таблицы в заданном поле Column\_Callback. |

* + 1. Получение исхода госпитализации по типу подразделения МО

**GET api/LeaveTypeByLpuUnitType**

|  |  |
| --- | --- |
| **Входящие параметры** | * RISHLeaveType\_id (N, О) – Исход госпитализации РИШ (федеральный справочник 1.2.643.5.1.13.2.1.1.542); * LpuUnitType\_id (N, Н) – Тип подразделения МО. |
| **Ответ** | Алгоритм следующий:  Применяется таблица стыковки dbo.RISHLeaveTypeLink.  Выходные параметры:   * RISHLeaveType\_id (N, О) – Исход госпитализации РИШ (федеральный справочник 1.2.643.5.1.13.2.1.1.542); * LpuUnitType\_id (N, Н) – Тип подразделения МО; * LeaveType\_id (N, О) – Исход госпитализации; * LeaveType\_fedid (N, Н) – Исход госпитализации федеральный (из справочника LeaveType). |

* + 1. Получение исхода заболевания по типу подразделения МО

**GET api/ResultDeseaseByLpuUnitType**

|  |  |
| --- | --- |
| **Входящие параметры** | * RISHResultDesease\_id (N, О) – Исход заболевания РИШ (федеральный справочник 1.2.643.5.1.13.2.1.1.688); * LpuUnitType\_id (N, Н) – Тип подразделения МО. |
| **Ответ** | Алгоритм следующий:  Применяется таблица стыковки dbo. RISHResultDeseaseLink.  Выходные параметры:   * RISHResultDesease\_id (N, О) – Исход заболевания РИШ (федеральный справочник 1.2.643.5.1.13.2.1.1.688); * LpuUnitType\_id (N, Н) – Тип подразделения МО; * ResultDesease\_id (N, О) – Исход заболевания. |

* + 1. Получение элементов справочника ЭРСБ с учетом таблицы стыковки

**GET api/ERSB/RefbookMap**

|  |  |
| --- | --- |
| **Входящие параметры** | * Refbook\_Code (T, О) – Код справочника; * Refbook\_MapName (T, Н) – Наименование таблицы стыковки; * Column\_Name (N, Н) – Наименование поля, по которому будет производиться фильтрация записей. Может принимать следующие значения:   + "id" - тогда фильтрация должна происходить по Ид справочника ЕЦП.   + "Code" - тогда фильтрация должна происходить по Коду справочника ЕЦП.   + "ERSB\_id" - тогда фильтрация должна происходить по Ид справочника ЭРСБ.   + "ERSB\_code" - тогда фильтрация должна происходить по Publ\_Code справочника ЭРСБ. * Column\_Value (T, Н) – Значение поля таблицы стыковки для фильтрации записей. |
| **Ответ** | Алгоритм следующий:   1. В таблице справочников dbo.ERSBRefbook по Коду справочника (Refbook\_Code) и наименованию таблицы стыковки (Refbook\_MapName, если параметр задан) определяется таблица стыковки. 2. Если таблица стыковки не найдена, то в ответе возвращается «0». 3. Иначе с помощью таблицы стыковки производится поиск записей, удовлетворяющих условию: Column\_Name = Column\_Value.   Если какой-то из двух параметров (Column\_Name, Column\_Value) не задан, то фильтрация не производится.  В ответе выводится список элементов со следующими полями:   * id – Идентификатор элемента справочника РИШ; * Code – Код элемента справочника РИШ; * ERSB\_id – Идентификатор справочника ЭРСБ; * ERSB\_Code – Код справочника ЭРСБ. |
| **Пример** |  |

* + 1. Получение справочника Статус очереди

**GET api/ EvnQueue/EvnQueueStatus**

|  |  |
| --- | --- |
| **Входящие параметры** | * Нет |
| **Действия** | Выдается список записей справочника Статус очереди (**EvnQueueStatus**) |
| **Ответ** | Если есть ошибки, то возвращается код ошибки.  Иначе: выдаются следующие данные:   * EvnQueueStatus\_id (N) – идентификатор статуса очереди. * EvnQueueStatus\_Name (Т) – наименование статуса очереди. * EvnQueueStatus\_Code (Т) – код статуса очереди. |
| **Пример** |  |

1. Паспорт медицинской организации
2. 1. Описание последовательности вызова методов по Структуре МО

В целях корректного сохранения данных в Системе должна быть соблюдена следующая последовательность вызова методов:

**Создание структуры МО**

МО добавляется в Систему по заявке от МО. Шаблон оформления заявки приведен в Приложении 1.

1. Создать подразделение.
2. Создать отделение.
3. Создать дополнительный профиль (при необходимости).
4. Создать участки.

Подробное описание методов см. 3.3.

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Описание метода** | **Метод** | **Комментарий** |
| 1 | Получение списка МО региона | GET api/LpuList |  |
| 2 | Создание подразделения МО | POST api/lpu/LpuBuilding | Выполняем метод 4. Если не найдено записей, то выполняем метод. Если найдена 1 запись, то выполняем метод 6. Если найдено несколько записей, то ошибка. |
| 3 | Получение атрибутов подразделения по идентификатору подразделения | GET api/lpu/LpuBuildingByID |  |
| 4 | Получение списка подразделений по коду и наименованию | GET api/lpu/LpuBuildingListByCodeAndName |  |
| 5 | Получение списка подразделений МО | GET api/lpu/LpuBuildingListByMO |  |
| 6 | Изменение данных подразделения. | PUT api/lpu/LpuBuilding |  |
| 7 | Создание отделения. | POST api/Lpu/LpuSection | В методе передаются данные группы отделений. При получении данных о группе отделений необходимо найти группу отделений или создать в случае если найти не удалось. Далее создавать отделения в найденной/созданной группе отделений.    Выполняем метод 8. Если не найдено записей, то выполняем метод. Если найдена 1 запись, то выполняем метод 12. Если найдено несколько записей, то ошибка. |
| 8 | Получение списка отделений по коду и наименованию | GET api/lpu/LpuSectionListByCodeAndName |  |
| 9 | Получение списка отделений в МО | GET api/Lpu/LpuSectionListByMO |  |
| 10 | Получение списка отделений в указанном подразделении | GET api/Lpu/LpuSectionListByBuilding |  |
| 11 | Получение данных отделения по идентификатору | GET api/Lpu/LpuSectionByid |  |
| 12 | Изменение данных отделения | PUT api/Lpu/LpuSection |  |
| 13 | Удаление отделения по идентификатору | DELETE api/Lpu/LpuSection |  |
| 14 | Создание дополнительного профиля отделения | POST api/Lpu/LpuSectionDopProfile | Выполняем метод 15. Если не найдено записей, то выполняем метод. Если найдена 1 запись, то выполняем метод 16. Если найдено несколько записей, то ошибка. |
| 15 | Получение списка дополнительных профилей отделения | GET api/Lpu/LpuSectionDopProfileList |  |
| 16 | Изменение дополнительного профиля. | PUT api/Lpu/LpuSectionDopProfile | Изменить можно только даты, профиль менять нельзя. По идентификатору отделения и коду профиля - поиск нужной записи. |
| 17 | Удаление дополнительного профиля | DELETE api/Lpu/LpuSectionDopProfile |  |
| 18 | Создание палаты в отделении | POST api/Lpu/LpuSectionWard | Доступно для отделений, которые входят в группу отделений с типом: Круглосуточный стационар, Дневной стационар при поликлинике, Дневной стационар при стационаре. Выполняем метод 19. Если не найдено записей, то выполняем метод. Если найдена 1 запись, то выполняем метод 20. Если найдено несколько записей, то ошибка. |
| 19 | Получение списка палат в отделении по наименованию | GET api/lpu/LpuSectionWardListByName |  |
| 20 | Изменение палаты. | PUT api/Lpu/LpuSectionWard |  |
| 21 | Удаление палаты по идентификатору | DELETE api/Lpu/LpuSectionWard |  |
| 22 | Список всех палат в отделении. | GET api/Lpu/LpuSectionWardListBySection | Доступно для отделений, которые входят в группу отделений с типом: Круглосуточный стационар, Дневной стационар при поликлинике, Дневной стационар при стационаре. |
| 23 | Получение атрибутов палаты по идентификатору. | GET api/Lpu/LpuSectionWardById |  |
| 24 | Добавление объектов комфортности в палате. | POST api/Lpu/LpuSectionWardComfortLink | Выполняем метод 25. Если не найдено записей, то выполняем метод. Если найдена 1 запись, то выполняем метод 26. Если найдено несколько записей, то ошибка. |
| 25 | Получение объектов комфортности в палате по наименованию | GET api/lpu/LpuSectionWardComfortLinkByName |  |
| 26 | Изменение объектов комфортности в палате | PUT api/Lpu/LpuSectionWardComfortLink |  |
| 27 | Получение списка объектов комфортности по палате | GET api/Lpu/LpuSectionWardComfotLinkListByWard |  |
| 28 | Удаление объектов комфортности по идентификатору | DELETE api/Lpu/LpuSectionWardComfortLink |  |
| 29 | Создание участка | POST api/Lpu/LpuRegion | Перед созданием участка необходимо проверить наличие места работы врача в отделении.  Выполняем метод 30. Если не найдено записей, то выполняем метод. Если найдена 1 запись, то выполняем метод 31. Если найдено несколько записей, то ошибка. |
| 30 | Получение списка участков по номеру | GET api/lpu/LpuRegionListByName |  |
| 31 | Редактирование участка | PUT api/Lpu/LpuRegion |  |
| 32 | Получение общих данных участка по идентификатору | PUT api/Lpu/LpuRegionByID |  |
| 33 | Получение списка участков по МО | PUT api/Lpu/LpuRegionListByМО |  |
| 34 | Создание периода работы врача на участке | POST api/Lpu/LpuRegionWorkerPlace | Выполняем метод 35. Если не найдено записей, то выполняем метод. Если найдена 1 запись, то выполняем метод 36. Если найдено несколько записей, то ошибка. |
| 35 | Получение информации о периоде работы врача по времени начала и окончания периода | GET api/Lpu/LpuRegionWorkerPlaceListByTime |  |
| 36 | Изменение периода работы врача на участке | PUT api/Lpu/LpuRegionWorkerPlace |  |
| 37 | Получение информации о периоде работы по идентификатору | GET api/Lpu/LpuRegionWorkerPlaceByID |  |
| 38 | Получение списка периодов работы врачей на участке | GET api/Lpu/LpuRegionWorkerPlaceListByRegion |  |
| 39 | Получение списка рабочих мест на участках по рабочему месту врача | GET api/Lpu/LpuRegionWorkerPlaceListByMedStaffFact |  |

* 1. Описание последовательности вызова методов по медицинскому персоналу

**Создание (добавление) сотрудника:**

Перед созданием сотрудника должна быть заведена Структура МО и создано штатное расписание.

1. Создать/найти человека.
2. Найти сотрудника, если не найден, то:
3. Создать сотрудника.

**Создание строки штатного расписания:**

Перед созданием строки штатного расписания должна быть заведена структура МО.

1. Найти должность, если не найдена, то:
2. Создать строку штатного расписания

**Создание места работы:**

Перед созданием места работы должна быть заведена структура МО, создано штатное расписание, создан сотрудник.

1. Создать/найти человека.
2. Найти сотрудника, если не найден, то:
3. Создать сотрудника
4. Найти должность, если не найдена, то:
5. Создать строку штатного расписания.
6. Создать место работы.
7. 2. 1. Описание последовательности вызова методов по созданию/получению человека

Создание и поиск человека выполняется методом GET api/Person, POST api/Person, описание см. **6.2**, **6.3**.

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Описание метода** | **Метод** | **Комментарий** |
| 1 | Поиск человека | GET api/Person | Если не найдено записей, то шаг Создание человека. Если найдена одна запись, то шаг Редактирование человека. Если найдено несколько записей, то ошибка "Найдено несколько пациентов по указанным параметрам. Просьба уточнить параметры." и выход. |
| 2 | Создание человека | POST api/Person | Если результат выполнения метода успешный, то в ответе возвращаются параметры:  - Person\_id и переход к Поиску полиса. Иначе ошибка "Ошибка выполнения метода "Создание человека" и выход. |
| 3 | Редактирование человека | PUT api/Person | Если результат выполнения метода успешный, то ответ "Успешно" и переход к шагу 4. Иначе предупреждение "Ошибка выполнения метода "Редактирование человека"" и переход к шагу 4. |
| 4 | Поиск полиса | GET api/Polis | Если не найдено записей, то шаг 5. Если найдена одна запись, то шаг 6. Если найдено несколько записей, то предупреждение "Найдено несколько полисов по указанным параметрам. Просьба уточнить параметры." и переход к шагу 7. |
| 5 | Создание полиса | POST api/Polis | Если результат выполнения метода успешный, то в ответе возвращаются параметры:  - Person\_id  - Polis\_id и переход к шагу 4. Иначе предупреждение "Ошибка выполнения метода "Создание полиса"" и переход к шагу 7. |
| 6 | Редактирование полиса | PUT api/Polis | Если результат выполнения метода успешный, то ответ "Успешно" и переход к шагу 7. Иначе предупреждение "Ошибка выполнения метода "Редактирование полиса"" и переход к шагу 7. |
| 7 | Поиск документа, удостоверяющего личность | GET api/Document | Если не найдено записей, то шаг 8. Если найдена одна запись, то шаг 9. Если найдено несколько записей, то предупреждение "Найдено несколько документов по указанным параметрам. Просьба уточнить параметры." и переход к шагу 10. |
| 8 | Создание документа, удостоверяющего личность | POST api/Document | Если результат выполнения метода успешный, то в ответе возвращаются параметры:  - Person\_id - Document\_id и переход к шагу 10. Иначе предупреждение "Ошибка выполнения метода "Создание документа, удостоверяющего личность"" и переход к шагу 10. |
| 9 | Редактирование документа, удостовреяющего личность | PUT api/Document | Если результат выполнения метода успешный, то ответ "Успешно" и переход к шагу 10. Иначе предупреждение "Ошибка выполнения метода "Редактирование документа, удостворяющего личность"" и переход к шагу 10. |
| 10 | Поиск адреса регистрации | GET api/Address | Если не найдено записей, то шаг 11. Если найдена одна запись, то шаг 12. Если найдено несколько записей, то предупреждение "Найдено несколько адресов по указанным параметрам. Просьба уточнить параметры." и переход к шагу 13. |
| 11 | Создание адреса регистрации | POST api/Address | Если результат выполнения метода успешный, то в ответе возвращаются параметры:  • Person\_id (N, Н) – Идентификатор человека; • Address\_id (N, О) – Идентификатор адреса; • AddressType\_id (N, Н) – Тип адреса (1 - Адрес регистрации, 2 - Адрес проживания, 3 - Адрес рождения) и переход к шагу 13. Иначе предупреждение "Ошибка выполнения метода "Создание адреса регистрации"" и переход к шагу 13. |
| 12 | Редактирование адреса регистрации | PUT api/Address | Если результат выполнения метода успешный, то ответ "Успешно" и переход к шагу 13. Иначе предупреждение "Ошибка выполнения метода "Редактирование Адреса регистрации"" и переход к шагу 13. |
| 13 | Поиск адреса проживания | GET api/Address | Если не найдено записей, то шаг 14. Если найдена одна запись, то шаг 15. Если найдено несколько записей, то предупреждение "Найдено несколько адресов по указанным параметрам. Просьба уточнить параметры." и переход к шагу 16. |
| 14 | Создание адреса проживания | POST api/Address | Если результат выполнения метода успешный, то в ответе возвращаются параметры:  • Person\_id (N, Н) – Идентификатор человека; • Address\_id (N, О) – Идентификатор адреса; • AddressType\_id (N, Н) – Тип адреса (1 - Адрес регистрации, 2 - Адрес проживания, 3 - Адрес рождения) и переход к шагу 16. Иначе предупреждение "Ошибка выполнения метода "Создание адреса проживания"" и переход к шагу 16. |
| 15 | Редактирование адреса проживания | PUT api/Address | Если результат выполнения метода успешный, то ответ "Успешно" и переход к шагу 16. Иначе предупреждение "Ошибка выполнения метода "Редактирование Адреса проживания"" и переход к шагу 16. |
| 16 | Поиск адреса рождения | GET api/Address | Если не найдено записей, то шаг 17. Если найдена одна запись, то шаг 18. Если найдено несколько записей, то предупреждение "Найдено несколько адресов по указанным параметрам. Просьба уточнить параметры." и выход. |
| 17 | Создание адреса рождения | POST api/Address | Если результат выполнения метода успешный, то в ответе возвращаются параметры:  • Person\_id (N, Н) – Идентификатор человека; • Address\_id (N, О) – Идентификатор адреса; • AddressType\_id (N, Н) – Тип адреса (1 - Адрес регистрации, 2 - Адрес проживания, 3 - Адрес рождения) и выход. Иначе предупреждение "Ошибка выполнения метода "Создание адреса рождения"" и выход. |
| 18 | Редактирование адреса рождения | PUT api/Address | Если результат выполнения метода успешный, то ответ "Успешно" и выход. Иначе предупреждение "Ошибка выполнения метода "Редактирование Адреса рождения"" и выход. |

* + 1. Описание последовательности вызова методов по созданию/получению сотрудника

Подробное описание методов см. 3.4.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Описание метода** | **Метод** | **Передаваемые параметры** | **Комментарий** |
| 1 | Поиск человека | GET api/Person | PersonSurName\_SurName, PersonFirName\_FirName, PersonBirthDay\_BirthDay, PersonSnils\_Snils | Если не найдено записей, то переходим к созданию человека. Если найдено несколько записей, то ошибка. |
| 2 | Поиск сотрудника | GET api/MedWorker | Person\_id | Если не найдено записей, то переходим к шагу 3. Если найдено несколько записей, то ошибка. |
| 3 | Создание сотрудника | POST api/MedWorker | Person\_id, CodeDLO |  |

* + 1. Описание последовательности вызова методов по созданию строки штатного расписания

Подробное описание методов см. 3.4.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Описание метода** | **Метод** | **Передаваемые параметры** | **Комментарий** |
| 1 | Поиск должности | GET api/PostByCode | Сode | Если не найдено записей, то должность создается через ЕЦП Если найдено несколько записей, то ошибка. |
| 2 | Создание строки штатного расписания | POST api/Staff | LpuSection\_id, Post\_id, isDummyStaff, PayType\_id, Rate, FRMPSubdivision\_id, BeginDate, EndDate, MedicalCareKind\_id, Сomments, RateNorm, LeadershipBonusPercent, SalaryReductionPercent, IsVillageBonus, SpecialLabourType, VacancyOfficialSalary, RateFinancing, UETCount |  |

* + 1. Описание последовательности вызова методов по созданию места работы

Подробное описание методов см. 3.4.

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Описание метода** | **Метод** | **Передаваемые параметры** |
| 1 | Создание места работы | POST api/MedStaffFact | MedPesonal\_id, Staff\_id, TabCode, IsDummyWP, PostOccupationType \_id, Rate, WorkMode\_id, MilitaryRelation\_id, FRMPSubdivision\_id, MedSpecOms\_id, Population, IsOMS, DisableWorkPlaceChooseInDocuments, Comments, RecType\_id, PriemTime, IsDirRec, IsQueueOnFree, IsNotReception, Descr, Contacts, DLOBeginDate, DLOEndDate, CommonLabourYears, CommonLabourMonths, CommonLabourDays, SpecialLabourDays, SpecialLabourMonths, SpecialLabourYears, ArriveRecordType\_id, ArriveOrderNumber, BeginDate, AdditionalAgreementDate, AdditionalAgreementNumber, LeaveRecordType\_id, LeaveOrderNumber, EndDate, OfficialSalary, QualificationLevel |

* 1. Описание методов работы со Структурой МО

  4. 1. Получение списка МО региона

**GET api/Lpu/LpuList**

Получение списка МО региона

|  |  |
| --- | --- |
| **Входящие параметры** | * Region\_id (N, О) - идентификатор региона, например, для Свердловской области передается «66» * Extended (N, H) – Опциональный параметр для вывода расширенных данных. * Если значение параметра «1», то в ответе появляются дополнительные поля – Address\_Address, Paddress\_Aaddress, Can\_Record, иначе указанные поля в ответе отстутствуют. |
| **Ответ** | Успешный ответ - список МО региона   * Lpu\_id (N,O) - Идентификатор МО * Org\_Name (T [300], O) - Наименование МО * Org\_Nick (T300], 0) - Краткое наименование ЛПУ * Address\_Address (T, H) - Юридический адрес МО и подразделений МО. Появляется в ответе, только если входящий параметр Extended = 1 * PAddress\_Address (T, H) - Фактический адрес МО и подразделений МО. Появляется в ответе, только если входящий параметр Extended = 1 * Can\_Record (N,H) – признак возможности записи в МО / подразделение МО.   Если одновременно выполняются условия:   * Не тестовая МО (Lpu\_IsTest принимает значения null или 1); * МО открыта на текущую дату; * Тип хотя бы одной группы отделений МО (LpuUnitType\_id) отличен от 1, 3, 6, 7, 9;   Хотя бы в одной из групп отделений, открытой на текущую дату:   * Установлен флаг «Включить запись операторами» (LpuUnit\_IsEnabled = 2); * Тип записи (RecType\_id) хотя бы одного, открытого на текущую дату, места работы (MedStaffFact) не равен null, 5, 6, то Can\_Record = 1, иначе 0. * **Регион – Пенза:** Если входящий параметр Extended = 1, то не отображаются МО, у которых на всех местах работы врачей установлен флаг «Не отображать на региональном портале». |

* + 1. Создание подразделения МО

**POST api/Lpu/LpuBuilding**

Создание подразделения МО.

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,O) - Идентификатор МО * LpuBuilding\_begDate (D, Н)  - Дата начала записи * LpuBuilding\_endDate (D, Н)  - Дата закрытия записи * LpuBuilding\_Code (T[50], O)  - Код подразделения. Должен быть уникальным в рамках МО. * LpuBuilding\_Name (T[100], O) - Наименование * LpuBuilding\_Nick  (T[30], H) - Краткое наименование * LpuBuildingType\_id (N[8], O) - Тип здания (значение справочника dbo.LpuBuildingType) * PAddress\_id (N[8], Н) - Адрес для выписки рецепта * Address\_id (N[8], Н) - Адрес подразделения * LpuBuilding\_Latitude (T[20], H) -  Широта * LpuBuilding\_Longitude (T[20], H) - Долгота * LpuBuilding\_RoutePlan (T[1000], H)  - Информация о маршруте проезда * LpuBuilding\_WorkTime (T[50], H) - Время работы подразделения * LpuBuilding\_IsExport (N[8], O) - Признак выгрузки подразделения. Возможные значения: 0 и 1, - где 0 – нет, 1 – да. |
| **Ответ** | Успешный ответ - идентификатор созданного подразделения МО   * LpuBuilding\_id (N[8],O) Идентификатор подразделения |

* + 1. Получение списка МО по району

**GET api/Lpu/LpuListBySubRgn**

Получение списка МО района

|  |  |
| --- | --- |
| **Входящие параметры** | * SubRgn\_id (N, О) - идентификатор района * Extended (N, H) – Опциональный параметр для вывода расширенных данных.   Если значение параметра «1», то в ответе появляются дополнительные поля – Address\_Address, Paddress\_Aaddress, Can\_Record, иначе указанные поля в ответе отстутствуют. |
| **Ответ** | Определяется список МО в территорию обслуживания, которых(OrgServiceTerr) входит указанный район.  Успешный ответ - список МО района   * Lpu\_id (N,O) - Идентификатор МО * Org\_Name (T [300], O) - Наименование МО * Org\_Nick (T300], 0) - Краткое наименование ЛПУ * Address\_Address (T, H) - Юридический адрес МО. Появляется в ответе, только если входящий параметр Extended = 1 * PAddress\_Address (T, H) - Фактический адрес МО. Появляется в ответе, только если входящий параметр Extended = 1 * Can\_Record (N,H) – признак возможности записи в МО / подразделение МО.   Если одновременно выполняются условия:   * Не тестовая МО (Lpu\_IsTest принимает значения null или 1); * МО открыта на текущую дату; * Тип хотя бы одной группы отделений МО (LpuUnitType\_id) отличен от 1, 3, 6, 7, 9;   Хотя бы в одной из групп отделений, открытой на текущую дату:   * Установлен флаг «Включить запись операторами» (LpuUnit\_IsEnabled = 2); * Тип записи (RecType\_id) хотя бы одного, открытого на текущую дату, места работы (MedStaffFact) не равен null, 5, 6, то Can\_Record = 1, иначе 0. |

* + 1. Получение атрибутов подразделения по идентификатору подразделения

**GET api/Lpu/LpuBuildingByID**

Получение атрибутов подразделения по идентификатору подразделения

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N[8], O)  - Идентификатор МО * LpuBuilding\_id (N[8],O) - Идентификатор подразделения |
| **Ответ** | Успешный ответ -  набор параметров подразделения   * Lpu\_id (N[8], O)  - Идентификатор МО * LpuBuilding\_id (N[8],O) - Идентификатор подразделения * LpuBuilding\_begDate (D, Н)  - Дата начала записи * LpuBuilding\_endDate (D, Н)  - Дата закрытия записи * LpuBuilding\_Code (T[50], O)  - Код подразделения * LpuBuilding\_Name (T[100], O) - Наименование * LpuBuilding\_Nick  (T[30], H) - Краткое наименование * LpuBuildingType\_id (N[8], O) - Тип здания (значение справочника dbo.LpuBuildingType) * PAddress\_id (N[8], Н) - Адрес для выписки рецепта * Address\_id (N[8], Н) - Адрес подразделения * LpuBuilding\_Latitude (T[20], H) -  Широта * LpuBuilding\_Longitude (T[20], H) - Долгота * LpuBuilding\_RoutePlan (T[1000], H)  - Информация о маршруте проезда * LpuBuilding\_WorkTime (T[50], H) - Время работы подразделения * LpuBuilding\_IsExport (N[8], H) - Признак выгрузки подразделения |

* + 1. Получение списка подразделений МО по коду и наименованию

**GET api/Lpu/LpuBuildingListByCodeAndName**

Получение списка подразделений по коду и наименованию

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N[8], O)  - Идентификатор МО * LpuBuilding\_Code (T[50], O)  - Код подразделения * LpuBuilding\_Name (T[100], H) - Наименование |
| **Ответ** | Успешный ответ - список подразделений МО   * LpuBuilding\_id (N[8],O) Идентификатор подразделения * LpuBuilding\_Code (T[50], O)  - Код подразделения * LpuBuilding\_Name (T[100], O) - Наименование * LpuBuildingType\_id (N[8], О) - Тип здания (значение справочника dbo.LpuBuildingType) |

* + 1. Изменение данных подразделения

**PUT api/Lpu/LpuBuilding**

Изменение данных подразделения. При выполнении метода происходит изменение указанных параметров.

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,O) - Идентификатор МО * LpuBuilding\_id (N[8], O)  - Идентификатор подразделения * LpuBuilding\_begDate (D, Н)  - Дата начала записи * LpuBuilding\_endDate (D, Н)  - Дата закрытия записи * LpuBuilding\_Code (T[50], Н)  - Код подразделения * LpuBuilding\_Name (T[100], Н) - Наименование * LpuBuilding\_Nick  (T[30], H) - Краткое наименование * LpuBuildingType\_id (N[8], Н) - Тип здания (значение справочника dbo.LpuBuildingType) * PAddress\_id (N[8], Н) - Адрес для выписки рецепта * Address\_id (N[8], Н) - Адрес подразделения * LpuBuilding\_Latitude (T[20], H) -  Широта * LpuBuilding\_Longitude (T[20], H) - Долгота * LpuBuilding\_RoutePlan (T[1000], H)  - Информация о маршруте проезда * LpuBuilding\_WorkTime (T[50], H) - Время работы подразделения * LpuBuilding\_IsExport (N[8], H) - Признак выгрузки подразделения |
| **Ответ** | **Успешный ответ**  Возвращается пустой ответ с кодом ошибки 0 |

* + 1. Создание отделения

**POST api/Lpu/LpuSection**

Создание отделения.

В методе передаются данные группы отделений. При получении данных о группе отделений необходимо найти группу отделений или создать, если найти не удалось. Далее создавать отделения в найденной/созданной группе отделений.

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuUnit\_id (N[8], O) – идентификатор группы отделений * LpuSection\_SetDate (D, O) - Дата создания * LpuSection\_DisDate (D, H) - Дата закрытия * LpuSectionProfile\_Code  (t[15], О) - Профиль отделения dbo.LpuSectionProfile * LpuSection\_Code (T[10], O) - Код отделения * LpuSectionCode\_id (N[8, У]) – идентификатор Справочника кодов отделений. * LpuSection\_Name (T[100], O) - Наименование отделения * MesAgeGroup\_id (N[8], H) - Возрастная группа * MESLevel\_id (N[8], H) - Уровень МЭС * LpuSection\_IsHTMedicalCare (N[8], O)  - Выполнять ВМП (только для стационарных отделений). Возможные значения: 0 и 1, где 0 – нет, 1 – да. * LpuSection\_KolAmbul - (N[8], H)  - Количество бригад скорой помощи * LpuSection\_KolJob - (N[8], H)  - Количество рабочих мест * LpuSection\_PlanAutopShift (N[8], H)-  Плановое число вскрытий в смену * LpuSection\_PlanResShift (N[8], H) - Плановое число исследований в смену * LpuSection\_PlanTrip (N[8], H)  - Плановое число выездов в смену * LpuSection\_PlanVisitDay (N[8], H)  - Плановое число посещений в сутки * LpuSection\_PlanVisitShift (N[8], H)  - Плановое число посещений в смену * LpuSectionAge\_id (N[8], H) - идентификатор типа отделения по возрасту (справочник dbo.LpuSectionAge) * LpuSectionBedProfile\_id - (N[8], H)  - идентификатор профиля коек ( справочник dbo.LpuSectionBedProfile); * LpuSectionOuter\_id (N, Н) – Идентификатор отделения МО в сторонней МИС. * FRMPSubdivision\_id (N[8], O) – код типа отделения (по ФРМР) для формы 30 (persis.FRMPSubdivision. persis.FRMPSubdivision\_id) * LpuBuildingPass\_id (N, Н) – Идентификатор здания МО.   **Выполняемые действия:**  **Проверки:**  – если не заполнены обязательные параметры, то ошибка.  – **Проверка обязательности заполнения Кода отделения:**  ***Регион Псков***: если не заполнен параметр Идентификтор справочника кодов отделений, то ошибка.  ***Другие регионы, кроме* *Псков***: если не заполнен Код отделения, то ошибка.  Если есть ошибки, то изменение данных не выполняется. Возвращаются коды ошибок.  Иначе: создается отделение в указанной группе отделений. |
| **Ответ** | **Успешный ответ**  Возвращает идентификатор созданного отделения   * LpuSection\_id (N[8], О) - идентификатор отделения |

* + 1. Получение списка отделений МО по коду и наименованию

**GET api/Lpu/LpuSectionListByCodeAndName**

Получение списка отделений по коду и наименованию

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N[8], O)  - Идентификатор МО * LpuSection\_Code (T[10], У) - Код отделения РМИС * LpuSectionCode\_Code (N[8, У]) – Код по Справочнику кодов отделений. * LpuSection\_Name (T[100], O) - Наименование отделения |
|  | Выполняются действия:  Выполняется проверка заполнения опбязательных полей, в т.ч.:  –**Проверка обязательности заполнения Кода отделения:**  ***Регион Псков***: если не заполнен параметр Код по справочнику кодов отделений, то ошибка.  ***Другие регионы, кроме* *Псков***: если не заполнен Код отделения РМИС, то ошибка.  Если ошибка, то возвращается код ошибки, действия не выполняются.  Действия:  – выполняется поиск отделений по заданным условиям.  – возвращается список отделений МО, удовлетворяющих заданным параметрам.  Если в структуре МО код отделения не указан и указан код отделения по справочнику кодов отделений, то в параметре код отделения указывается код по справочнику кодов отделений. |
| **Ответ** | Если ошибка, то возвращается код ошибки.  Успешный ответ - список отделений МО   * LpuSection\_id (N[8], О) - идентификатор отделения * LS.LpuBuilding\_id (N[8], О) – идентификатор здания в структуре МО. * LpuUnit\_id (N[8], O) – идентификатор группы отделений * LpuUnit\_Code (T[5], O) - Код группы отделений * LpuUnitType\_Code (N[4], О) - Тип группы отделений, код из справочника  [dbo.LpuUnitType](http://fixuploader.swn.local/dbspr/index/1968830176) * LpuUnit\_Name (T[50], O) - Наименование группы отделений * LpuSection\_setDate (D, O) - Дата создания * LpuSection\_disDate (D, H) - Дата закрытия * LpuSection\_Code (T[10], У) - Код отделения * LpuSectionProfile\_Code  (T[15], О) - Профиль отделения dbo.LpuSectionProfile * LpuSectionCode\_id (N[8, У]) – идентификатор Справочника кодов отделений. * LpuSection\_Name (T[100], O) - Наименование отделения; * LpuSectionOuter\_id (N, Н) – Идентификатор отделения МО в сторонеей МИС. |

* + 1. Получение списка отделений в МО

**GET api/Lpu/LpuSectionListByMO**

Список всех отделений, находящихся в МО

|  |  |
| --- | --- |
| **Входящие параметры** | Обязательно должен быть указан один из параметров: Lpu\_id, Lpu\_OID.   * Lpu\_id (N[8], Н)  - Идентификатор МО * Lpu\_OID (T[50], Н) (**Регион: Базовый, кроме Казахстана**) – OID МО в ФРМО. |
| **Ответ** | **Успешный ответ**  Возвращает список отделений   * LpuSection\_id (N[8], О) - идентификатор отделения * Lpu\_id (N[8], О) – идентификатор МО, к которой относится отделение * LpuUnit\_id (N[8], O) – идентификатор группы отделений * LpuUnit\_Code (T[5], O) - Код группы отделений * LpuUnitType\_Code (N[4], О) - Тип группы отделений, код из справочника  dbo.LpuUnitType * LpuUnit\_Name (T[50], O) - Наименование группы отделений * LpuSection\_setDate (D, O) - Дата создания * LpuSection\_disDate (D, H) - Дата закрытия * LpuSectionProfile\_id (N[25], О) – идентификатор основного профиля отделения * LpuSectionProfile\_Code  (T[15], О) – код основного профиля отделения dbo.LpuSectionProfile * LpuSection\_Code (T[10], У) - Код отделения * LpuSectionCode\_id (N[8, У]) – идентификатор Справочника кодов отделений. * LpuSection\_Name (T[100], O) - Наименование отделения; * LpuSectionOuter\_id (N, Н) – Идентификатор отделения МО в сторонеей МИС. |

* + 1. Получение списка отделений в указанном подразделении

**GET api/Lpu/LpuSectionListByBuilding**

Список отделений находящихся в указанном подразделении.

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuBuilding\_id (N[8], О)  - Идентификатор подразделения |
| **Ответ** | **Успешный ответ**  Возвращает список отделений подразделения   * LpuSection\_id (N[8], О) - идентификатор отделения * LpuUnit\_id (N[8], O) – идентификатор группы отделений * LpuUnit\_Code (T[5], O) - Код группы отделений * LpuUnitType\_Code (N[4], О) - Тип группы отделений, код из справочника  dbo.LpuUnitType * LpuUnit\_Name (T[50], O) - Наименование группы отделений * LpuSection\_SetDate (D, O) - Дата создания * LpuSection\_DisDate (D, H) - Дата закрытия * LpuSectionProfile\_Code  (S[15], О) - Профиль отделения dbo.LpuSectionProfile * LpuSection\_Code (T[10], O) - Код отделения * LpuSectionCode\_id (N[8, У]) – идентификатор Справочника кодов отделений. * LpuSection\_Name (T[100], O) - Наименование отделения; * LpuSectionOuter\_id (N, Н) – Идентификатор отделения МО в сторонеей МИС. |

* + 1. Получение данных отделения по идентификатору

**GET api/Lpu/LpuSectionById**

Данные отделения по идентификатору отделения

|  |  |
| --- | --- |
| **Входящие параметры** | Обязательно должен быть указан один из параметров: LpuSection\_id, LpuSectionOuter\_id, LpuSection\_OID.   * LpuSection\_id (N, H) - Идентификатор отделения; * LpuSectionOuter\_id (N, Н) – Идентификатор отделения МО в сторонней МИС. * LpuSection\_OID (T[100], Н) (**Регион: *Базовый*, кроме *Казахстана*** ) – OID отделения в справочнике отделений и кабинетов ФРМО. |
| **Ответ** | Если не передан ни один параметр, то ошибка.  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – атрибуты отделения:   * LpuSection\_setDate (D, O) - Дата создания * LpuSection\_disDate (D, H) - Дата закрытия * Lpu\_id (N,O) - идентификатор МО, к которой относится отделение * LpuSectionProfile\_id (N, О) – Ид профиля отделения; * LpuSectionProfile\_Code  (T[15], О) - Код профиля отделения dbo.LpuSectionProfile * LpuUnit\_id (N[8], O) – идентификатор группы отделений * LpuSection\_Code (T[10], У) - Код отделения * LpuSectionCode\_id (N[8, У]) – идентификатор Справочника кодов отделений. * LpuSection\_Name (T[100], O) - Наименование отделения * MesAgeGroup\_id (N[8], H) - Возрастная группа * MESLevel\_id (N[8], H) - Уровень МЭС * LpuSection\_IsHTMedicalCare (N[8], H)  - Выполнять ВМП (только для стационарных отделений) * LpuSection\_KolAmbul - (N[8], H)  - Количество бригад скорой помощи * LpuSection\_KolJob - (N[8], H)  - Количество рабочих мест * LpuSection\_PlanAutopShift (N[8], H)-  Плановое число вскрытий в смену * LpuSection\_PlanResShift (N[8], H) - Плановое число исследований в смену * LpuSection\_PlanTrip (N[8], H)  - Плановое число выездов в смену * LpuSection\_PlanVisitDay (N[8], H)  - Плановое число посещений в сутки * LpuSection\_PlanVisitShift (N[8], H)  - Плановое число посещений в смену * LpuSectionAge\_id (N[8], H) - идентификатор типа отделения по возрасту (справочник dbo.LpuSectionAge) * LpuSectionBedProfile\_id - (N[8], H)  - идентификатор профиля коек ( справочник dbo.LpuSectionBedProfile); * LpuUnitType\_id (N, Н) – Ид типа группы отделений. * FRMPSubdivision\_id (N[8], O) – код типа отделения (по ФРМР) для формы 30 (persis.FRMPSubdivision. persis.FRMPSubdivision\_id) * LpuBuildingPass\_id (N, Н) – Идентификатор здания МО |

* + 1. Изменение данных отделения

**PUT api/Lpu/LpuSection**

Изменение данных отделения

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuSection\_id (N[8], О) - идентификатор отделения * LpuSection\_setDate (D, H) - Дата создания * LpuSection\_disDate (D, H) - Дата закрытия * LpuSectionProfile\_Code  (T[15], H) - Профиль отделения dbo.LpuSectionProfile * LpuUnit\_id (N[8], O) – идентификатор группы отделений * LpuSection\_Code (T[10], H) - Код отделения * LpuSectionCode\_id (N[8, У]) – идентификатор Справочника кодов отделений. * LpuSection\_Name (T[100], H) - Наименование отделения * MesAgeGroup\_id (N[8], H) - Возрастная группа * MESLevel\_id (N[8], H) - Уровень МЭС * LpuSection\_IsHTMedicalCare (N[8], H)  - Выполнять ВМП (только для стационарных отделений) * LpuSection\_KolAmbul - (N[8], H)  - Количество бригад скорой помощи * LpuSection\_KolJob - (N[8], H)  - Количество рабочих мест * LpuSection\_PlanAutopShift (N[8], H)-  Плановое число вскрытий в смену * LpuSection\_PlanResShift (N[8], H) - Плановое число исследований в смену * LpuSection\_PlanTrip (N[8], H)  - Плановое число выездов в смену * LpuSection\_PlanVisitDay (N[8], H)  - Плановое число посещений в сутки * LpuSection\_PlanVisitShift (N[8], H)  - Плановое число посещений в смену * LpuSectionAge\_id (N[8], H) - идентификатор типа отделения по возрасту (справочник dbo.LpuSectionAge) * LpuSectionBedProfile\_id - (N[8], H)  - идентификатор профиля коек ( справочник dbo.LpuSectionBedProfile); * LpuSectionOuter\_id (N, Н) – Идентификатор отделения МО в сторонней МИС. * FRMPSubdivision\_id (N[8], O) – код типа отделения (по ФРМР) для формы 30 (persis.FRMPSubdivision. persis.FRMPSubdivision\_id) * LpuBuildingPass\_id (N, Н) – Идентификатор здания МО |
|  | Выполняемые действия:  **Проверки:**  Выполняется проверка заполнения опбязательных полей, в т.ч.:  ***Регион Псков***: Проверка обязательности заполнения Кода отделения: если не заполнен параметр Код по справочнику кодов отделений, то ошибка.  Если есть ошибки, то изменение данных не выполняется. Возвращаются коды ошибок.  Иначе:  - выполняется изменение данных отделения. |
| **Ответ** | **Успешный ответ**  Возвращается пустой ответ с кодом ошибки 0 |

* + 1. Удаление отделения по идентификатору

**DELETE api/Lpu/LpuSection**

Удаление отделения по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuSection\_id (N[8], О) - идентификатор отделения |
| **Ответ** | **Успешный ответ**  Возвращается пустой ответ с кодом ошибки 0 |

* + 1. Создание дополнительного профиля

**POST api/Lpu/LpuSectionDopProfile**

Создание дополнительного профиля отделения

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuSection\_id (N[8], О) - идентификатор отделения * LpuSectionProfile\_id  (N[8], О) – профиль отделения, значение из справочника dbo.LpuSectionProfile * LpuSectionLpuSectionProfile\_begDate (D, O) - Дата начала * LpuSectionLpuSectionProfile\_endDate (D, H) - Дата окончания |
| **Ответ** | **Успешный ответ**   * LpuSectionLpuSectionProfile\_id (N, O) – Идентификатор дополнительного профиля   **Ошибки**  Нельзя добавить дополнительный профиль, если он является основным.  Дата начала действия доп. профиля не может быть раньше даты создания отделения. |

* + 1. Получение списка дополнительных профилей отделения

**GET api/Lpu/LpuSectionDopProfileList**

Получение списка дополнительных профилей отделения

|  |  |
| --- | --- |
| **Входящие параметры** | Обязательно должен быть указан один из параметров: LpuSection\_id, LpuSection\_OID.   * LpuSection\_id (N[8], Н) - идентификатор отделения * LpuSection\_OID (T[100], Н) (**Регион: *Базовый*, кроме *Казахстана***) – OID отделения в справочнике отделений и кабинетов ФРМО. |
| **Ответ** | **Успешный ответ**  Возвращает массив, сордержащий список дополнительных профилей с атрибутами: {   * LpuSectionLpuSectionProfile\_id (N, O) – Идентификатор дополнительного профиля * LpuSectionProfile\_Code – Код доп. профиля * LpuSectionLpuSectionProfile\_begDate (D, O) - Дата начала * LpuSectionLpuSectionProfile\_endDate (D, H) - Дата окончания } |

* + 1. Изменение дополнительного профиля

**PUT api/Lpu/LpuSectionDopProfile**

Изменение дополнительного профиля. Изменить можно только даты, профиль менять нельзя. Дата начала действия доп. профиля не может быть раньше даты создания отделения. По идентификатору отделения и коду профиля - поиск нужной записи.

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuSectionProfile\_id  (N[8], H) – Идентиифкатор профиля отделения. Значение справочника dbo.LpuSectionProfile * LpuSectionLpuSectionProfile\_id (N, О) – Идентификатор дополнительного профиля * LpuSectionProfile\_code  (T[15], H) - Профиль отделения dbo.LpuSectionProfile * LpuSectionLpuSectionProfile\_begDate (D, H) - Дата начала * LpuSectionLpuSectionProfile\_endDate (D, H) - Дата окончания |
| **Ответ** | **Успешный ответ**  Возвращается пустой ответ с кодом ошибки 0  **Ошибки**  Если не удалось найти дополнительный профиль, то ошибка |

* + 1. Удаление дополнительного профиля

**DELETE api/Lpu/LpuSectionDopProfile**

Удаление дополнительного профиля

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuSection\_id (N[8], О) - идентификатор отделения * LpuSectionLpuSectionProfile\_id (N[8], О) – Идентиифкатор профиля отделения dbo.LpuSectionProfile |
| **Ответ** | **Успешный ответ**  Возвращается пустой ответ с кодом ошибки 0  **Ошибки**  Если не удалось найти дополнительный профиль, то ошибка. Если Lpu\_id пользователя не совпадает с Lpu\_id отделения, из которого удаляется доп. профиль, то ошибка. |

* + 1. Создание палаты в отделении

**POST api/Lpu/LpuSectionWard**

Создание палаты. Доступно для отделений, которые входят в группу отделений с типом: Круглосуточный стационар, Дневной стационар при поликлинике, Дневной стационар при стационаре.

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuSection\_id (N[8], О) - идентификатор отделения * LpuSectionWard\_Name (T, O) – Наименование (номер) палаты * LpuWardType\_id (N, O) – Тип палаты. Значения из справочника dbo.LpuWardType * Sex\_id (N, O) – Вид палаты. Значения из справочника dbo.Sex * LpuSectionWard\_MainPlace (N, H) – Количество основных мест в палате. * LpuSectionWard\_DopPlace (N, H) – Количество дополнительных мест в палате. * LpuSectionWard\_BedRepair (N, H) – Количество коек на ремонте. * LpuSectionWard\_Square (N, H) – Площадь палаты. * LpuSectionWard\_DayCost (N, H) – Стоимость нахождения в сутки. Формат денег: с разделителями и 2 знаками после запятой. * LpuSectionWard\_Views (T[256,H]) – Вид из окна * LpuSectionWard\_setDate (D, O) – дата начала периода действия * LpuSectionWard\_disDate (D, H) – дата окончания периода действия |
| **Ответ** | **Успешный ответ**   * LpuWardType\_id (N[8], О) - идентификатор палаты |

* + 1. Получение списка палат отделения по наименованию

**GET api/Lpu/LpuSectionWardListByName**

Получение списка палат в отделении по наименованию

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuSection\_id (N[8], О) - Идентификатор отделения * LpuSectionWard\_Name (T, O) – Наименование (номер) палаты |
| **Ответ** | Успешный ответ - список палат отделения   * LpuSectionWard\_id (N[8], О) - идентификатор палаты |

* + 1. Изменение палаты

**PUT api/Lpu/LpuSectionWard**

Изменение атрибутов палаты.

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuSectionWard\_id (N[8], О) - идентификатор палаты * LpuSectionWard\_Name (T, H) – Наименование (номер) палаты * LpuWardType\_id (N, H) – Тип палаты. Значения из справочника dbo.LpuWardType * Sex\_id (N, H) – Вид палаты. Значения из справочника dbo.Sex * LpuSectionWard\_MainPlace (N, H) – Количество основных мест в палате. * LpuSectionWard\_DopPlace (N, H) – Количество дополнительных мест в палате. * LpuSectionWard\_BedRepair (N, H) – Количество коек на ремонте. * LpuSectionWard\_Square (N, H) – Площадь палаты. * LpuSectionWard\_DayCost (N, H) – Стоимость нахождения в сутки. Формат денег: с разделителями и 2 знаками после запятой. * LpuSectionWard\_Views (T[256,H]) – Вид из окна * LpuSectionWard\_setDate (D, H) – дата начала периода действия * LpuSectionWard\_disDate (D, H) – дата окончания периода действия |
| **Ответ** | **Успешный ответ**   * Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Удаление палаты по идентификатору

**DELETE api/Lpu/LpuSectionWard**

Удаление палаты по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuSectionWard\_id (N[8], О) - идентификатор палаты |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Получение списка палат в отделении

**GET api/Lpu/LpuSectionWardListBySection**

Список всех палат в отделении. Доступно для отделений, которые входят в группу отделений с типом: Круглосуточный стационар, Дневной стационар при поликлинике, Дневной стационар при стационаре.

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuSection\_id (N, O)  - Идентификатор отделения |
| **Ответ** | **Успешный ответ**  Возвращает список палат:   * LpuSectionWard\_id (N,О) - идентификатор палаты |

* + 1. Получение атрибутов палаты по идентификатору

**GET api/Lpu/LpuSectionWardById**

Получение атрибутов палаты по идентификатору.

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuSectionWard\_id (N[8], О) - идентификатор палаты |
| **Ответ** | **Успешный ответ**   * LpuSectionWard\_Name (T, O) – Наименование (номер) палаты * LpuWardType\_id (N, O) – Тип палаты. * Sex\_id (N, O) – Вид палаты. * LpuSectionWard\_MainPlace (N, H) – Количество основных мест в палате. * LpuSectionWard\_DopPlace (N, H) – Количество дополнительных мест в палате. * LpuSectionWard\_BedRepair (N, H) – Количество коек на ремонте. * LpuSectionWard\_Square (N, H) – Площадь палаты. * LpuSectionWard\_DayCost (N, H) – Стоимость нахождения в сутки. Формат денег: с разделителями и 2 знаками после запятой. * LpuSectionWard\_Views (T[256,H]) – Вид из окна * LpuSectionWard\_setDate (D, O) – дата начала периода действия * LpuSectionWard\_disDate (D, H) – дата окончания периода действия |

* + 1. Добавление объектов комфортности в палате

**POST api/Lpu/LpuSectionWardComfortLink**

Добавление объектов комфортности в палату.

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuSectionWard\_id (N,О) - идентификатор палаты. * LpuSectionWardComfortLink\_Count (N, O) – количество объектов. * DChamberComfort\_id (N, O) – наименование объекта. Значение из справочника passport.DChamberComfort |
| **Ответ** | **Успешный ответ**   * LpuSectionWardComfortLink\_id (N, O) – Идентификатор объектов комфорта в палате |

* + 1. Получение объектов комфортности палаты по наименованию

**GET api/Lpu/LpuSectionWardComfortLinkByName**

Получение объектов комфортности в палате по наименованию

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuSectionWard\_id (N,О) - идентификатор палаты. * DChamberComfort\_id (N, O) – наименование объекта. Значение из справочника passport.DChamberComfort |
| **Ответ** | Успешный ответ:   * LpuSectionWardComfortLink\_id (N, O) – идентификатор объектов комфорта в палате |

* + 1. Изменение объектов комфортности в палате

**PUT api/Lpu/LpuSectionWardComfortLink**

Изменение объектов комфортности в палате

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuSectionWardComfortLink\_id (N, O) – идентификатор объектов комфорта в палате * LpuSectionWardComfortLink\_Count (N, H) – количество объектов. * DChamberComfort\_id (N, H) – наименование объекта. Значение из справочника passport.DChamberComfort |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Получение списка объектов комфортности по палате

**GET api/Lpu/LpuSectionWardComfotLinkListByWard**

Получение списка объектов комфортности по палате

|  |  |
| --- | --- |
| **Входящие параметры** | Список идентификаторов объектов комфорта в палате:   * LpuSectionWard\_id (N, O) – идентификатор палаты |
| **Ответ** | **Успешный ответ:**  Возвращается cписок объектов комфортности:   * LpuSectionWardComfortLink\_id (N, O) – Идентификатор объектов комфорта в палате |

* + 1. Удаление объектов комфортности по идентификатору

**DELETE api/Lpu/LpuSectionWardComfortLink**

Удаление объектов комфортности по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuSectionWardComfortLink\_id (N, O) – Идентификатор объектов комфорта в палате |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Создание участка

**POST api/Lpu/LpuRegion**

Создание участка.

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id - (N[8], О) - идентификатор МО * LpuSection\_id (N[8], О) - идентификатор отделения * LpuBuilding\_id (N[8], О) - идентификатор подразделения * LpuRegionType\_id (N[8], О) – тип участка. Справочник LpuRegionType * LpuRegion\_Name (Т[30, O]) – номер участка * LpuRegion\_begDate (D, O) – дата создания * LpuRegion\_endDate (D, H) – дата закрытия |
| **Ответ** | **Успешный ответ**   * LpuRegion\_id (N[8], О) - идентификатор участка   **Ошибки**  Если не удалось создать участок, то ошибка |

* + 1. Получение списка участков по номеру

**GET api/Lpu/LpuRegionListByName**

Получение списка участков по номеру

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuBuilding\_id (N[8], О) - идентификатор подразделения * LpuSection\_id (N[8], О) - Идентификатор отделения * LpuRegion\_Name (Т[30, O]) – номер участка |
| **Ответ** | Успешный ответ - список участков   * LpuRegion\_id (N[8], О) - идентификатор участка |

* + 1. Редактирование участка

**PUT api/Lpu/LpuRegion**

Редактирование данных участков

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuRegion\_id (N[8], О) - идентификатор участка * LpuSection\_id (N[8], О) - идентификатор отделения * LpuBuilding\_id (N[8], О) - идентификатор подразделения * LpuRegion\_Name (Т[30, H]) – номер участка * LpuRegion\_begDate (D, H) – дата создания * LpuRegion\_endDate (D, H) – дата закрытия |
| **Ответ** | **Успешный ответ**  Возвращается пустой ответ с кодом ошибки 0 |

* + 1. Получение общих данных участка по идентификатору

**GET api/Lpu/LpuRegionByID**

Получение общих данных участка по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuRegion\_id (N[8], О) - идентификатор участка |
| **Ответ** | **Успешный ответ**   * Lpu\_id - (N[8], О) - идентификатор МО * LpuSection\_id (N[8], О) - идентификатор отделения * LpuBuilding\_id (N[8], О) - идентификатор подразделения * LpuRegionType\_id (N[8], О) – тип участка. Справочник LpuRegionType * LpuRegion\_Name (Т[30, O]) – номер участка * LpuRegion\_begDate (D, O) – дата создания * LpuRegion\_endDate (D, H) – дата закрытия |

* + 1. Получение списка участков по МО

**GET api/Lpu/LpuRegionListByМО**

Получение списка участков по МО

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id - (N[8], О) - идентификатор МО |
| **Ответ** | **Успешный ответ список участков**   * Lpu\_id - (N[8], О) - идентификатор МО * LpuSection\_id (N[8], О) - идентификатор отделения * LpuBuilding\_id (N[8], О) - идентификатор подразделения * LpuRegionType\_id (N[8], О) – тип участка. Справочник LpuRegionType * LpuRegion\_Name (Т[30, O]) – номер участка * LpuRegion\_begDate (D, O) – дата создания * LpuRegion\_endDate (D, H) – дата закрытия |

* + 1. Создание периода работы врача на участке

**POST**  **api/Lpu/LpuRegionWorkerPlace**

Создание периода работы врача на участке

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuRegion \_id - (N[8], О) - идентификатор участка * MedStaffFact\_id (N, O) – идентификатор места работы * MedStaffRegion\_begDate - (D, O) – дата начала периода * MedStaffRegion\_endDate - (D, H) – дата окончания периода   MedStaffRegion\_isMain (N[8], O) – признак основного врача на участке Возможные значения: 0 и 1, - где 0 – нет, 1 – да |
| **Ответ** | **Успешный ответ идентификатор записи**   * MedStaffRegion\_id (N[8], О) – идентификатор периода работы врача на участке |

* + 1. Изменение периода работы врача на участке

**PUT**  **api/Lpu/LpuRegionWorkerPlace**

Изменение периода работы врача на участке

|  |  |
| --- | --- |
| **Входящие параметры** | * MedStaffRegion\_id (N[8], О) – идентификатор периода работы врача на участке * MedStaffFact\_id (N, H) – идентификатор места работы * MedStaffRegion\_begDate - (D, H) – дата начала периода * MedStaffRegion\_endDate - (D, H) – дата окончания периода * MedStaffRegion\_isMain (N[8], Н) – признак основного врача на участке |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0 |

* + 1. Получение информации о периоде работы врача на участке по датам начала и окончания

**GET**  **api/Lpu/LpuRegionWorkerPlaceListByTime**

Получение информации о периоде работы врача по времени начала и окончания периода

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuRegion \_id - (N[8], О) - идентификатор участка * MedStaffFact\_id (N, O) – идентификатор места работы * MedStaffRegion\_begDate - (D, O) – дата начала периода * MedStaffRegion\_endDate - (D, H) – дата окончания периода |
| **Ответ** | **Успешный ответ**  Список периодов работы врача:   * MedStaffRegion\_id (N[8], О) – идентификатор периода работы врача на участке |

* + 1. Получение информации о периоде работы по идентификатору

**GET**  **api/Lpu/LpuRegionWorkerPlaceByID**

Получение информации о периоде работы врача по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * MedStaffRegion\_id (N[8], О) – идентификатор периода работы врача на участке |
| **Ответ** | Успешный ответ – данные по периоду работы врача на участке:   * MedStaffRegion\_begDate - (D, O) – дата начала периода * MedStaffRegion\_endDate - (D, H) – дата окончания периода * MedStaffRegion\_isMain (N[8], Н) – признак основного врача на участке |

* + 1. Получение периодов работы врачей на участке

**GET**  **api/Lpu/LpuRegionWorkerPlaceListByRegion**

Получение списка периодов работы врачей на участке

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuRegion \_id - (N[8], О) - идентификатор участка |
| **Ответ** | **Успешный ответ:**  Список периодов работы врачей на участке:   * MedStaffRegion\_id (N[8], О) – идентификатор периода работы врача на участке |

* + 1. Получение списка периодов работы врачей по рабочему месту врача

**GET**  **api/Lpu/LpuRegionWorkerPlaceListByMedStaffFact**

Получение списка рабочих мест на участках по рабочему месту врача

|  |  |
| --- | --- |
| **Входящие параметры** | * MedStaffFact\_id - (N[8], О) – рабочее место |
| **Ответ** | **Успешный ответ:**  Список периодов работы врача на участке:   * MedStaffRegion\_id (N[8], О) – идентификатор периода работы врача на участке |

* + 1. Создание группы отделений МО

**POST**  **api/Lpu/LpuUnit**

Создание группы отделений МО

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuBuilding\_id - (N[8],O) - идентификатор подразделения * LpuUnitType\_id (N, О) – Тип подразделения МО * LpuUnitTypeDop\_id (N, Н) - дополнительный тип отделений * Address\_id (N[8], Н) – адрес группы отделений * LpuUnit\_Code (T[5], О)- код группы отделений * LpuUnit\_Name (T[100], О) – наименование группы отеделений * LpuUnit\_Descr (T[200], Н) – примечание по записи на прием. * LpuUnit\_Phone (T[100], Н) – телефон записи на прием. * LpuUnit\_IsEnabled (N[8], Н)- признак активности * LpuUnit\_IsDirWithRec (N[8], Н)- признак записи * LpuUnit\_ExtMedCnt (N[8], Н)- признак приема по скорой помощи. * LpuUnit\_Email (T[50], Н)- Электронная почта * LpuUnit\_IP (T[100], Н) - Адрес и способ подключения * LpuUnitSet\_id (N[8], Н) - идентификатор обьединения групп отделений * LpuUnit\_Guid (T[36], Н) – ГУИД * LpuUnit\_begDate (D, Н)- дата начала действия записи * LpuUnit\_endDate (D, Н)- дата окончания действия записи * LpuUnit\_IsOMS (N[8], Н) - Признак Работает по ОМС * UnitDepartType\_fid (N[8], O) - Тип (ФРМО) * LpuUnitProfile\_fid (N[8], O) - Профиль (ФРМО) * LpuUnit\_isStandalone (N[8], О) - Признак "Обособленное подразделение" * LpuBuildingPass\_id (N[8], Н) - Основное здание * LpuUnit\_isHomeVisit (N[8], Н) - Признак "Приём на дому" * LpuUnit\_isCMP (N[8], Н) - Признак "Приём скорой помощи" * LpuUnit\_FRMOUnitID (N[8], Н) - идентификатор ФРМО * LpuUnit\_FRMOid (N[8], Н) – идентификатор ФРМО для структурного подразделения |
| **Выполняемые действия** | Проверки:  – если не заполнены обязательные параметры, то ошибка.  – выполняется проверка полученных входящих параметров, значения которых заданы идентификаторами .Если в соответствующих объектах РМИС указанного значения нет, то ошибка.  Если есть ошибки, то создание данных не выполняется. Возвращаются коды ошибок.  Иначе: создается группа отделений. |
| **Ответ** | **Если есть ошибки, то коды ошибок**  **Иначе:** LpuUnit \_id (N[8], О) – идентификатор группы отделений |
| **Пример** | Запрос  Ответ |

* + 1. Изменение группы отделений МО

**PUT**  **api/Lpu/LpuUnit**

Изменение группы отделений МО

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuUnit\_id - (N[8],O) - идентификатор группы отделений * LpuUnitType\_id (N, О) – Тип подразделения МО * LpuUnitTypeDop\_id (N, Н) - дополнительный тип отделений * Address\_id (N[8], Н) – адрес группы отделений * LpuUnit\_Code (T[5], О)- код группы отделений * LpuUnit\_Name (T[100], О) – наименование группы отеделений * LpuUnit\_Descr (T[200], Н) – Примечание по записи на прием. * LpuUnit\_Phone (T[100], Н) – телефон записи на прием. * LpuUnit\_isEnabled (N[8], Н)- признак активности * LpuUnit\_isDirWithRec (N[8], Н)- признак записи * LpuUnit\_ExtMedCnt (N[8], Н)- признак приема по скорой помощи. * LpuUnit\_Email (T[50], Н)- Электронная почта * LpuUnit\_IP (T[100], Н) - Адрес и способ подключения * LpuUnitSet\_id (N[8], Н) - идентификатор обьединения групп отделений * LpuUnit\_Guid (T[36], Н) – ГУИД * LpuUnit\_begDate (D, Н)- дата начала действия записи * LpuUnit\_endDate (D, Н)- дата окончания действия записи * LpuUnit\_isOMS (N[8], Н) - Признак Работает по ОМС * UnitDepartType\_fid (N[8], O) - Тип (ФРМО) * LpuUnitProfile\_fid (N[8], O) - Профиль (ФРМО) * LpuUnit\_isStandalone (N[8], О) - Признак "Обособленное подразделение" * LpuBuildingPass\_id (N[8], Н) - Основное здание * LpuUnit\_isHomeVisit (N[8], Н) - Признак "Приём на дому" * LpuUnit\_isCMP (N[8], Н) - Признак "Приём скорой помощи" * LpuUnit\_FRMOUnitID (N[8], Н) - идентификатор ФРМО * LpuUnit\_FRMOid (N[8], Н) – идентификатор ФРМО для структурного подразделения |
| **Действия** | Проверки:  – если не заполнены обязательные параметры, то ошибка.  – выполняется проверка полученных входящих параметров, значения которых заданы идентификаторами .Если в соответствующих объектах РМИС указанного значения нет, то ошибка.  Если есть ошибки, то изменение данных не выполняется. Возвращаются коды ошибок.  Иначе: даныне изменяются. |
| **Ответ** | **Если есть ошибки, то коды ошибок**  **Иначе:** LpuUnit \_id (N[8], О) – идентификатор группы отделений |
| **Пример** | Запрос  Ответ |

* + 1. Получение данных о группе отделений МО по идентификатору

**GET**  **api/Lpu/LpuUnitById**

Получение группы отделений МО по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | Обязательно должен быть указан один из параметров: LpuUnit\_id, LpuUnit\_OID.   * LpuUnit\_id (N[8], Н) – идентификатор группы отделения * LpuUnit\_OID (T[100], Н) (**Регион: *Базовый*, кроме *Казахстана***) – OID структурного подразделения в справочнике структурных подразделений ФРМО. |
| **Ответ** | * LpuBuilding\_id - (N[8],O) - идентификатор подразделения * LpuUnitType\_id (N, Н) – Тип подразделения МО * LpuUnitTypeDop\_id (N, Н) - дополнительный тип отделений * Address\_id (N[8], Н) – адрес группы отделений * LpuUnit\_Code (T[5], Н)- код группы отделений * LpuUnit\_Name (T[100], Н) – наименование группы отеделений * LpuUnit\_Descr (T[200], Н) – примечание по записи на прием. * LpuUnit\_Phone (T[100], Н) – телефон записи на прием. * LpuUnit\_isEnabled (N[8], Н)- признак активности * LpuUnit\_isDirWithRec (N[8], Н)- признак записи * LpuUnit\_ExtMedCnt (N[8], Н)- признак приема по скорой помощи. * LpuUnit\_Email (T[50], Н)- Электронная почта * LpuUnit\_IP (T[100], Н) - Адрес и способ подключения * LpuUnitSet\_id (N[8], Н) - идентификатор обьединения групп отделений * LpuUnit\_Guid (T[36], Н) – ГУИД * LpuUnit\_begDate (D, Н)- дата начала действия записи * LpuUnit\_endDate (D, Н)- дата окончания действия записи * LpuUnit\_isOMS (N[8], Н) - Признак Работает по ОМС * UnitDepartType\_fid (N[8], O) - Тип (ФРМО) * LpuUnitProfile\_fid (N[8], O) - Профиль (ФРМО) * LpuUnit\_isStandalone (N[8], Н) - Признак "Обособленное подразделение" * LpuBuildingPass\_id (N[8], Н) - Основное здание * LpuUnit\_isHomeVisit (N[8], Н) - Признак "Приём на дому" * LpuUnit\_isCMP (N[8], Н) -- Признак "Приём скорой помощи" * LpuUnit\_FRMOUnitID (N[8], Н) - идентификатор ФРМО * LpuUnit\_FRMOid (N[8], Н) – идентификатор ФРМО для структурного подразделения |
| **Пример** | Запрос  Ответ |

* + 1. Получение списка групп отделений МО по идентификатору подразделения

**GET**  **api/Lpu/LpuUnitList**

Получение списка группы отделений МО по идентификатору подразделения

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuBuilding\_id - (N[8],O) - идентификатор подразделения |
| **Ответ** | * LpuUnit\_id (N[8], Н) – идентификатор группы отделения   Возвращается список идентификаторов групп отделений МО указанного подразделения |
| **Пример** | Запрос  Ответ: |

* + 1. Создание коек в Коечном фонде отделения

**POST**  **api/Lpu/LpuUnit/LpuSectionBedState**

Создание в коек в коечном фонде отделения

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuSection\_id – (N[8],O) – идентификатор отделения ЛПУ * LpuSectionBedProfile\_id – (N[8],O) – профиль коек * LpuSectionBedState\_ProfileName – (T[200],Н) – Наименование * LpuSectionBedState\_Fact– (N[8],O) –фактическое количество * LpuSectionBedState\_Plan– (N[8], O) – планируемое количество * LpuSectionBedState\_Repair– (N[8], O) – отклонение от планируемого количества * LpuSectionBedState\_CountOms– (N[8],Н) –Количество коек, оплачиваемых по ОМС * LpuSectionBedState\_MalePlan– (N[8],Н) –Мужские койки. Плановое количество * LpuSectionBedState\_MaleFact– (N[8],Н) –Мужские койки Фактическое количество * LpuSectionBedState\_FemalePlan– (N[8],Н) –Женские койки. Плановое количество * LpuSectionBedState\_FemaleFact– (N[8],Н) –Женские койки. Фактическое количество * LpuSectionBedState\_begDate – (D,О) – дата начала * LpuSectionBedState\_endDate – (D,Н) – дата окончания * LpuSectionBedProfileLink\_fedid – (N[8],O) – федеральный профиль коек |
| **Выполняемые действия** | Проверки:  – если не заполнены обязательные параметры, то ошибка.  – выполняется проверка на наличие коек с таким же профилем, у которых период действия имеет пересечение с периодом действия коек, переданных методом. Если есть пересечение, то ошибка: Есть койки с таким же профилем, период действия которых пересекается с указанным.  Если есть ошибки, то создание данных не выполняется. Возвращаются коды ошибок.  Иначе: Выполняется создание данных. |
| **Ответ** | **Если есть ошибки, то возвращаются коды ошибок.**  **Иначе: возвращается идентификатор созданной записи**   * LpuSectionBedState\_id – (N[8], О) – идентификатор коек и их профилей в отделении МО |

* + 1. Изменение коек в коечном фонде отделения

**PUT**  **api/Lpu/LpuUnit/LpuSectionBedState**

Изменение коек в коечном фонде отделения

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuSectionBedState\_id – (N[8], О) – идентификатор коек и их профилей в отделении МО * LpuSectionBedProfile\_id – (N[8],O) – профиль коек * LpuSectionBedState\_ProfileName – (T[200],Н) – Наименование * LpuSectionBedState\_Fact– (N[8],O) –фактическое количество * LpuSectionBedState\_Plan– (N[8],О) – планируемое количество * LpuSectionBedState\_Repair– (N[8],О) – отклонение от планируемого количества * LpuSectionBedState\_CountOms– (N[8],Н) –Количество коек, оплачиваемых по ОМС * LpuSectionBedState\_MalePlan– (N[8],Н) –Мужские койки. Плановое количество * LpuSectionBedState\_MaleFact– (N[8],Н) –Мужские койки Фактическое количество * LpuSectionBedState\_FemalePlan– (N[8],Н) –Женские койки. Плановое количество * LpuSectionBedState\_FemaleFact– (N[8],Н) –Женские койки. Фактическое количество * LpuSectionBedState\_begDate – (D,О) – дата начала * LpuSectionBedState\_endDate – (D,Н) – дата окончания * LpuSectionBedProfileLink\_fedid – (N[8],O) – федеральный профиль коек |
| **Выполняемые действия** | Проверки:  – если не заполнены обязательные параметры, то ошибка.  – выполняется проверка на наличие коек с таким же профилем, у которых период действия имеет пересечение с периодом действия коек, переданных методом. Если есть пересечение, то ошибка: Есть койки с таким же профилем, период действия которых пересекается с указанным.  Если есть ошибки, то изменение данных не выполняется. Возвращаются коды ошибок.  Иначе: выполняется изменение данных. |
| **Ответ** | Возвращаются коды ошибок.  При отсутствии ошибок возвращается код 0. |

* + 1. Удаление в отделении МО коек по профилю

**DELETE api/Lpu/LpuUnit/LpuSectionBedState**

Удаление в отделении МО коек по профилю

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuSectionBedState\_id – (N[8], О) – идентификатор коек и их профилей в отделении МО |
| **Выполняемые действия** | Если есть койки с таким идентификатором, то выполняется удаление данных .  Иначе: ошибка. |
| **Ответ** | Возвращаются коды ошибок.  При отсутствии ошибок возвращается код 0. |
| **Пример** | Запрос  Ответ |

* + 1. Получение списка коек коечного фонда отделения МО

**GET api/Lpu/LpuUnit/LpuSectionBedStateListBySection**

Получение списка коек коечного фонда отделения МО

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuSection\_id – (N[8], О) – идентификатор отделения МО |
| **Выполняемые действия** | Возвращается список коек отделения МО (из **LpuSectionBedState**) |
| **Исходящие параметры** | * LpuSectionBedState\_id – (N[8], О) – идентификатор коек и их профилей в отделении МО * LpuSectionBedProfile\_id – (N[8],O) – профиль коек * LpuSectionBedState\_ProfileName – (T[200],Н) – Наименование * LpuSectionBedState\_Fact– (N[8],O) –фактическое количество * LpuSectionBedState\_Plan– (N[8], O) – планируемое количество * LpuSectionBedState\_Repair– (N[8], O) – отклонение от планируемого количества * LpuSectionBedState\_CountOms– (N[8],Н) –Количество коек, оплачиваемых по ОМС * LpuSectionBedState\_MalePlan– (N[8],Н) –Мужские койки. Плановое количество * LpuSectionBedState\_MaleFact– (N[8],Н) –Мужские койки Фактическое количество * LpuSectionBedState\_FemalePlan– (N[8],Н) –Женские койки. Плановое количество * LpuSectionBedState\_FemaleFact– (N[8],Н) –Женские койки. Фактическое количество * LpuSectionBedState\_begDate – (D,О) – дата начала * LpuSectionBedState\_endDate – (D,Н) – дата окончания * LpuSectionBedProfileLink\_fedid – (N[8],O) – федеральный профиль коек |

* + 1. Получение списка служб по МО

**GET api/Lpu/MedServiceByMO**

Получение списка служб по МО

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,O) - Идентификатор МО * MedService\_IsExternal (N,Н) – Признак внешней службы (2- внешняя служба, 1 или null – не внешняя служба) |
| **Ответ** | Успешный ответ - список служб   * MedService\_id (N,O) – идентификатор службы * MedService\_IsExternal (N,O) – признак внешней службы * MedService\_Name (Т,О) – наименование * MedService\_Nick (Т,Н) – краткое наименование * MedServiceType\_id (N,O) – тип служб |

* 1. Описание методов работы с сотрудниками МО
  2. 1. Создание сотрудника

**POST api/MedWorker**

Создание сотрудника.

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, О) – идентификатор пациента; * CodeDLO (S, Н) - код ДЛО; * HonouredBrevetDate (D, Н) - Дата получения почетного звания 'Заслуженный врач Российской Федерации'; * PeoplesBrevetDate (D, Н) - Дата получения почетного звания 'Народный врач СССР'; |
| **Ответ** | **Успешный ответ**  Возвращает идентификатор сотрудника   * MedWorker\_id (N, О) – идентификатор сотрудника |

* + 1. Получение данных сотрудника по идентификатору человека

**GET api/MedWorker**

Получения сотрудника по человеку

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, О) – идентификатор пациента; |
| **Ответ** | **Успешный ответ**  Возвращает идентификатор сотрудника   * MedWorker\_id (N, О) – идентификатор сотрудника |

* + 1. Получение данных сотрудника по идентификатору сотрудника

**GET api/MedWorkerById**

Получение сотрудника по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * MedWorker\_id (N, О) – идентификатор сотрудника |
| **Ответ** | **Успешный ответ**   * Person\_id (N, О) – идентификатор пациента; * CodeDLO (S, Н) - код ДЛО; * HonouredBrevetDate (D, Н) - Дата получения почетного звания 'Заслуженный врач Российской Федерации'; * PeoplesBrevetDate (D, Н) - Дата получения почетного звания 'Народный врач СССР'; |

* + 1. Получение мест работы медицинского работника (на данный момент)

**GET api/MedStaffFactByMedPersonal**

|  |  |
| --- | --- |
| **Запрос** | * MedPersonal\_id (N, O) – Идентификатор сотрудника; * Staff\_id (N, H) – Идентификатор строки штатного расписания; * MedSpecOms\_id(N, H) – Идентификатор специальности врача; * Post\_id (N, H) – Идентификатор должности врача; * LpuUnit\_id (N, H) – Идентификатор группы отделений; * LpuSection\_id (N, H) – Идентификатор отделения; * LpuUnitType\_id (N, H) – Идентификатор типа группы отделений МО; * LpuSectionProfile\_id (N, H) – Идентификатор профиля отделения МО; |
| **Ответ** | Успешный ответ – список мест работы:   * MedStaffFact\_id (N, O) – Идентификатор места работы врача. * PersonSurName\_SurName (T[50], O) – Фамилия врача; * PersonFirName\_FirName (T[50], O) – Имя врача; * PersonSecName\_SecName (T[50], Н) – Отчество врача; |
| **Пример** |  |

* + 1. Получение данных о местах работы сотрудника

**GET api/WorkPlace**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N,Н) – Ид человека; * MedWorker\_id (N, Н) – Ид сотрудника. |
| **Ответ** | Производится проверка: хотя бы один из параметров Person\_id или MedWorker\_id должен быть заполнен. Иначе ошибка.  В ответе возвращается записи со следующими параметрами:   * MedWorker\_id (N, О) – Ид сотрудника; * Person\_id (N, О) – Ид человека; * Lpu\_id (N, О) – МО; * LpuSection\_id (N, Н) – Отделение МО; * PostOccupationType\_id (N, О) – Тип занятия должности * Post\_id (N, О) – Должность; * Rate (N, О) – Ставка; * begiDate (D, О) – Дата начала; * endDate (D, Н) – Дата окончания; * LeaveRecordType\_id (N, Н) – Тип окончания записи; * LeaveRecordDelType\_id (N, Н) – Причина выбытия; * LpuUnit\_FRMO (N, Н) – Ид группы отделений из сервиса ФРМО; * LpuSection\_FRMO (N, Н) – Ид Отделения стационара из сервиса ФРМО; * DismissalReason\_id (N, Н) – Причина увольнения. |
| **Пример** |  |

* + 1. Создание данных о среднем или профессиональном образовании сотрудника

**POST api/SpecialityDiploma**

Создание данных о среднем или профессиональном образовании сотрудника

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N,Н) – Ид человека; * MedWorker\_id (N, Н) – Ид сотрудника; * YearOfGraduation (N, О) – Год окончания; * DiplomaNumber (T, О) – Номер диплома; * DiplomaSeries (T, О) – Серия диплома; * OtherEducationalInstitution (T, Н) – Иное учебное заведение; * DiplomaSpeciality\_id (N, О) – Специальность; * EducationType\_id (N, О) – Тип образования; * EducationInstitution\_id (N, Н) – Ид учебного заведения; * IsSpecSet (N, Н) – Признак «Целевой набор»; * FRMPTerritories\_id (N, Н) – Субъект РФ; * KLCountry\_id (N, Н) – Ид страны учебного заведения (из справочника учебных заведений); * YearOfAdmission (N, О) – Год поступления; * DocumentRecieveDate (N, О) – Дата выдачи диплома; * Qualification\_id (N, Н) – Квалификация сотрудника |
| **Выполняются действия** | Производится проверка:  - хотя бы один из параметров Person\_id или MedWorker\_id должен быть заполнен. Иначе ошибка. Данные не создаются.  - значения, полученные в виде идентификаторов должны быть определены в соответствующих им таблицах РМИС. Если значения не определены, то ошибка.  Если есть ошибки, то данные не создаются. Возвращается код ошики.  Иначе: создаются данные о среднем или профессиональном образовании сотрудника. |
| **Ответ** | Если есть ошибки, то возвращается код ошибки.  Иначе: в ответе возвращается идентификатор записи:   * id (N, О) – Ид записи о среднем/профессиональном образовании; |
| **Пример** |  |

* + 1. Изменение данных о среднем или профессиональном образовании сотрудника

**PUT api/SpecialityDiploma**

Изменение данных о среднем или профессиональном образовании сотрудника

|  |  |
| --- | --- |
| **Входящие параметры** | * id (N, О) – Ид записи о среднем/профессиональном образовании; * YearOfGraduation (N, О) – Год окончания; * DiplomaNumber (T, О) – Номер диплома; * DiplomaSeries (T, О) – Серия диплома; * OtherEducationalInstitution (T, Н) – Иное учебное заведение; * DiplomaSpeciality\_id (N, О) – Специальность; * EducationType\_id (N, О) – Тип образования; * EducationInstitution\_id (N, Н) – Ид учебного заведения; * IsSpecSet (N, Н) – Признак «Целевой набор»; * FRMPTerritories\_id (N, Н) – Субъект РФ; * KLCountry\_id (N, Н) – Ид страны учебного заведения (из справочника учебных заведений); * YearOfAdmission (N, О) – Год поступления; * DocumentRecieveDate (N, О) – Дата выдачи диплома; * Qualification\_id (N, Н) – Квалификация сотрудника |
| **Выполняются действия** | Производится проверка:  - если указанный идентификтор записи об образовании отсутствует, то ошибка..  - значения, полученные в виде идентификаторов должны быть определены в соответствующих им таблицах РМИС. Если значения не определены, то ошибка.  Если есть ошибки, то данные не изменяются. Возвращается код ошики.  Иначе: изменяются данные о среднем или профессиональном образовании сотрудника. |
| **Ответ** | Ответ: если ошибка, то возвращается код ошибки.  Если ошибок нет, то возвращается код – 0. |
| **Пример** |  |

* + 1. Удаление данных о среднем или профессиональном образовании сотрудника

**DELETE api/SpecialityDiploma**

Удаление данных о среднем или профессиональном образовании сотрудника

|  |  |
| --- | --- |
| **Входящие параметры** | * id (N, О) – Ид записи о среднем/профессиональном образовании; |
| **Выполняются действия** | Проверка: если указанный идентификтор записи об образовании отсутствует, то удвление не выполняется и возвращается ошибка.  Иначе: данные удаляются. |
| **Ответ** | Если есть ошибка, то возвращается код ошибки.  Иначе: возвращается код – 0. |
| **Пример** |  |

* + 1. Получение данных о среднем или профессиональном образовании сотрудника

**GET api/SpecialityDiploma**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N,Н) – Ид человека; * MedWorker\_id (N, Н) – Ид сотрудника; * EducationType\_id (S, Н) – Тип образования. Допускается указывать несколько Ид. В этом случае Ид указываются через запятую. Использование пробелов допустимо. Пример: во входном параметре EducationType\_id допустимо передавать следующие значения: «1», «2,3», «2, 3». |
| **Ответ** | Производится проверка: хотя бы один из параметров Person\_id или MedWorker\_id должен быть заполнен. Иначе ошибка.  В ответе возвращается записи со следующими параметрами:   * id (N, О) – Ид записи о профессиональном образовании; * MedWorker\_id (N, О) – Ид сотрудника; * Person\_id (N, О) – Ид человека; * YearOfGraduation (N, О) – Год окончания; * DiplomaNumber (T, О) – Номер диплома; * DiplomaSeries (T, О) – Серия диплома; * OtherEducationalInstitution (T, Н) – Иное учебное заведение; * DiplomaSpeciality\_id (N, О) – Специальность; * EducationType\_id (N, О) – Тип образования; * EducationInstitution\_id (N, Н) – Ид учебного заведения; * IsSpecSet (N, Н) – Признак «Целевой набор»; * FRMPTerritories\_id (N, Н) – Субъект РФ; * KLCountry\_id (N, Н) – Ид страны учебного заведения (из справочника учебных заведений); * YearOfAdmission (N, О) – Год поступления; * DocumentRecieveDate (N, О) – Дата выдачи диплома; * Qualification\_id (N, Н) – Квалификация сотрудника. |
| **Пример** |  |

* + 1. Создание данных о послевузовском образовании сотрудника

**POST api/PostgraduateEducation**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N,Н) – Ид человека; * MedWorker\_id (N, Н) – Ид сотрудника. * graduationDate (D, О) – Дата получения диплома; * startDate (D, О) – Дата начала обучения; * DiplomaNumber (T, О) – Номер диплома; * endDate (D, О) – Дата окончания обучения; * DiplomaSeries (T, Н) – Серия диплома; * PostgraduateEducationType\_id (N, О) – Тип образования; * OtherEducationalInstitution (T, Н) – Иное учебное заведение; * AcademicMedicalDegree\_id (N, Н) – Ученая степень; * Speciality\_id (N, Н) – Специальность при обучении в интернатуре, ординатуре; * EducationInstitution\_id (N, Н) – Учебное заведение; * IsSpecSet (N, Н) – Признак «Целевой набор»; * FRMPTerritories\_id (N, Н) – Субъект РФ; * SpecialityAspirant\_id (N, Н) – Специальность при обучении в аспирантуре и докторантуре. |
| **Выполняются действия** | Проверка:  - если хотя бы один из параметров Person\_id или MedWorker\_id должен быть заполнен. Иначе ошибка.  - значения идентификаторов, полученных во входящих параметрах, дожны быть определены в соответствующих объектах РМИС. Если хотя бы одного из значений нет, то ошибка.  Если ошибка, то данные не создаются. Возвращается код ошибки.  Иначе: Создаются данные о послевузовском образовании сотрудника |
| **Ответ** | Если есть ошибки, тов возвращается код ошибки.  Иначе: возвращается записи со следующими параметрами:   * id (N, О) – Ид записи о послевузовском образовании;. |
| **Пример** |  |

* + 1. Изменение данных о послевузовском образовании сотрудника

**PUT api/PostgraduateEducation**

|  |  |
| --- | --- |
| **Входящие параметры** | * id (N, О) – Ид записи о послевузовском образовании;. * graduationDate (D, О) – Дата получения диплома; * startDate (D, О) – Дата начала обучения; * DiplomaNumber (T, О) – Номер диплома; * endDate (D, О) – Дата окончания обучения; * DiplomaSeries (T, Н) – Серия диплома; * PostgraduateEducationType\_id (N, О) – Тип образования; * OtherEducationalInstitution (T, Н) – Иное учебное заведение; * AcademicMedicalDegree\_id (N, Н) – Ученая степень; * Speciality\_id (N, Н) – Специальность при обучении в интернатуре, ординатуре; * EducationInstitution\_id (N, Н) – Учебное заведение; * IsSpecSet (N, Н) – Признак «Целевой набор»; * FRMPTerritories\_id (N, Н) – Субъект РФ; * SpecialityAspirant\_id (N, Н) – Специальность при обучении в аспирантуре и докторантуре. |
| **Выполняются действия** | Проверка:  - если указанный идентификатор образования не найден, то ошибка  - значения идентификаторов, полученных во входящих параметрах, дожны быть определены в соответствующих объектах РМИС. Если хотя бы одного из значений нет, то ошибка.  Если ошибка, то данные не изменяются. Возвращается код ошибки.  Иначе: изменяются данные о послевузовском образовании сотрудника. |
| **Ответ** | Если при выполнении метода были ошибки, то возвращается код ошибки.  Если данные изменены, то возвращается код 0. |
| **Пример** |  |

* + 1. Удаление данных о послевузовском образовании сотрудника

**DELETE api/PostgraduateEducation**

|  |  |
| --- | --- |
| **Входящие параметры** | * id (N, О) – Ид записи о послевузовском образовании;. |
| **Выполняются действия** | Производится проверка: если указанный идентификатор образования не найден, то ошибка. Данные не удаляются.  Иначе: Указанная запись удаляется. |
| **Ответ** | Если при выполнении метода были ошибки, то возвращается код ошибки.  Если данные изменены, то возвращается код 0. |
| **Пример** |  |

* + 1. Получение данных о послевузовском образовании сотрудника

**GET api/PostgraduateEducation**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N,Н) – Ид человека; * MedWorker\_id (N, Н) – Ид сотрудника. |
| **Ответ** | Производится проверка: хотя бы один из параметров Person\_id или MedWorker\_id должен быть заполнен. Иначе ошибка.  В ответе возвращается записи со следующими параметрами:   * id (N, О) – Ид записи о послевузовском образовании; * MedWorker\_id (N, О) – Ид сотрудника; * Person\_id (N, О) – Ид человека; * graduationDate (D, О) – Дата получения диплома; * startDate (D, О) – Дата начала обучения; * DiplomaNumber (T, О) – Номер диплома; * endDate (D, О) – Дата окончания обучения; * DiplomaSeries (T, Н) – Серия диплома; * PostgraduateEducationType\_id (N, О) – Тип образования; * OtherEducationalInstitution (T, Н) – Иное учебное заведение; * AcademicMedicalDegree\_id (N, Н) – Ученая степень; * Speciality\_id (N, Н) – Специальность при обучении в интернатуре, ординатуре; * EducationInstitution\_id (N, Н) – Учебное заведение; * IsSpecSet (N, Н) – Признак «Целевой набор»; * FRMPTerritories\_id (N, Н) – Субъект РФ; * SpecialityAspirant\_id (N, Н) – Специальность при обучении в аспирантуре и докторнатуре. |
| **Пример** |  |

* + 1. Создание данных о курсах переподготовки сотрудника

**POST api/RetrainingCourse**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N,Н) – Ид человека; * MedWorker\_id (N, Н) – Ид сотрудника. * PassYear (N, О) – Год прохождения; * HoursCount (N, О) – Количество часов; * DocumentNumber (T, Н) – Номер документа; * DocumentSeries (T, Н) – Серия документа; * OtherEducationalInstitution (T, Н) – Иное учебное заведение; * Speciality\_id (N, О) – Специальность; * EducationInstitution\_id (N, Н) – Учебное заведение; * DocumentRecieveDate (N, О) – Дата выдачи диплома |
| **Выполняются действия** | Проверки:   * Должен быть указан хотя бы один из параметров Person\_id или MedWorker\_id должен быть заполнен. Иначе ошибка. * Если не найден указанный человек или медсотрудник, то ошибка. * Если не заполнены обязательные параметры, то ошибка.   Если есть ошибки, то возвращается код ошибки.  Иначе: создаются данные о курсах переподготовки сотрудника. |
| **Ответ** | В ответе возвращается:  Если есть ошибки, то код ошибки.  Иначе: записи со следующими параметрами:   * id (N, О) – Ид записи о курсе переподготовки; |
| **Пример** |  |

* + 1. Изменение данных о курсах переподготовки сотрудника

**PUT api/RetrainingCourse**

|  |  |
| --- | --- |
| **Входящие параметры** | * id (N, О) – Ид записи о курсе переподготовки; * PassYear (N, О) – Год прохождения; * HoursCount (N, О) – Количество часов; * DocumentNumber (T, Н) – Номер документа; * DocumentSeries (T, Н) – Серия документа; * OtherEducationalInstitution (T, Н) – Иное учебное заведение; * Speciality\_id (N, О) – Специальность; * EducationInstitution\_id (N, Н) – Учебное заведение; * DocumentRecieveDate (N, О) – Дата выдачи диплома |
| **Выполняются действия** | Производятся проверки:   * Если не заполнены обязательные параметры, то ошибка. * Выполняется проверка полученных входящих параметров, значения которых являются идентификаторами. Если в соответствующих объектах РМИС значения для них не определены, то ошибка.   Если есть ошибки, то возвращается код ошибки.  Иначе: изменяются данные о курсах переподготовки сотрудника. |
| **Ответ** | Если при выполнении метода были ошибки, то возвращается код ошибки.  Иначе: возвращается код 0. |
| **Пример** |  |

* + 1. Удаление данных о курсах переподготовки сотрудника

**DELETE api/RetrainingCourse**

|  |  |
| --- | --- |
| **Входящие параметры** | * id (N, О) – Ид записи о курсе переподготовки; |
| **Выполняются действия** | Производится проверка: если указанный идентификатор курсов переподготовки не найден, то ошибка. Данные не удаляются.  Иначе: Указанная запись удаляется. |
| **Ответ** | Если при выполнении метода были ошибки, то возвращается код ошибки.  Если данные удалены, то возвращается код 0. |

* + 1. Получение данных о курсах переподготовки сотрудника

**GET api/RetrainingCourse**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N,Н) – Ид человека; * MedWorker\_id (N, Н) – Ид сотрудника. |
| **Ответ** | Производится проверка: хотя бы один из параметров Person\_id или MedWorker\_id должен быть заполнен. Иначе ошибка.  В ответе возвращается записи со следующими параметрами:   * id (N, О) – Ид записи о курсе переподготовки; * MedWorker\_id (N, О) – Ид сотрудника; * Person\_id (N, О) – Ид человека; * PassYear (N, О) – Год прохождения; * HoursCount (N, О) – Количество часов; * DocumentNumber (T, Н) – Номер документа; * DocumentSeries (T, Н) – Серия документа; * OtherEducationalInstitution (T, Н) – Иное учебное заведение; * Speciality\_id (N, О) – Специальность; * EducationInstitution\_id (N, Н) – Учебное заведение; * DocumentRecieveDate (N, О) – Дата выдачи диплома. |
| **Пример** |  |

* + 1. Создание данных о курсах повышения квалификации

**POST api/QualificationImprovementCourse**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N,Н) – Ид человека; * MedWorker\_id (N, Н) – Ид сотрудника. * Year (N, О) – Год прохождения; * DocumentRecieveDate (D, О) – Дата получения документа; * HoursCount (N, О) – Количество часов; * DocumentNumber (T, О) – Номер документа; * DocumentSeries (T, Н) – Серия документа; * OtherEducationalInstitution (T, Н) – Иное учебное заведение; * Round (T, О) – Цикл; * Speciality\_id (N, О) – Специальность; * EducationInstitution\_id (N, Н) – Учебное заведение. |
| **Выполняются действия** | Производятся проверки:   * Должен быть указан хотя бы один из параметров Person\_id или MedWorker\_id должен быть заполнен. Иначе ошибка. * Если не заполнены обязательные параметры, то ошибка. * Выполняется проверка полученных входящих параметров, значения которых заданы идентификаторами. Если в соответствующих объектах РМИС эти значения не определены, то ошибка.   Если есть ошибки, то возвращается код ошибки.  Иначе: создаются данные о курсах повышения квалификации. |
| **Ответ** | * id (N, О) – Ид записи о курсе повышения квалификации; |
| **Пример** |  |

* + 1. Изменение данных о курсах повышения квалификации

**PUT api/QualificationImprovementCourse**

|  |  |
| --- | --- |
| **Входящие параметры** | * id (N, О) – Ид записи о курсе повышения квалификации; * Year (N, О) – Год прохождения; * DocumentRecieveDate (D, О) – Дата получения документа; * HoursCount (N, О) – Количество часов; * DocumentNumber (T, О) – Номер документа; * DocumentSeries (T, Н) – Серия документа; * OtherEducationalInstitution (T, Н) – Иное учебное заведение; * Round (T, О) – Цикл; * Speciality\_id (N, О) – Специальность; * EducationInstitution\_id (N, Н) – Учебное заведение. |
| **Выполняются действия** | Производятся проверки:   * Если не заполнены обязательные параметры, то ошибка. * Выполняется проверка полученных входящих параметров, значения которых заданы идентификаторами. Если в соответствующих объектах РМИС эти значения не определены, то ошибка.   Если есть ошибки, то возвращается код ошибки.  Иначе: изменяются данные о курсах повышения квалификации. |
| **Ответ** | Если при выполнении метода были ошибки, то возвращается код ошибки.  Иначе: возвращается код 0. |
| **Пример** |  |

* + 1. Удаление данных о курсах повышения квалификации

**DELETE api/QualificationImprovementCourse**

|  |  |
| --- | --- |
| **Входящие параметры** | * id (N, О) – Ид записи о курсе переподготовки; |
| **Выполняются действия** | Производится проверка: если указанный идентификатор курсов повышения квалификации не найден, то ошибка. Данные не удяляются. Возвращается код ошибки.  Иначе: Указанная запись удаляется. |
| **Ответ** | Если при выполнении метода были ошибки, то возвращается код ошибки.  Иначе: возвращается код 0. |

* + 1. Получение данных о курсах повышения квалификации

**GET api/QualificationImprovementCourse**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N,Н) – Ид человека; * MedWorker\_id (N, Н) – Ид сотрудника. |
| **Ответ** | Производится проверка: хотя бы один из параметров Person\_id или MedWorker\_id должен быть заполнен. Иначе ошибка.  В ответе возвращается записи со следующими параметрами:   * id (N, О) – Ид записи о курсе повышения квалификации; * MedWorker\_id (N, О) – Ид сотрудника; * Person\_id (N, О) – Ид человека; * Year (N, О) – Год прохождения; * DocumentRecieveDate (D, О) – Дата получения документа; * HoursCount (N, О) – Количество часов; * DocumentNumber (T, О) – Номер документа; * DocumentSeries (T, Н) – Серия документа; * OtherEducationalInstitution (T, Н) – Иное учебное заведение; * Round (T, О) – Цикл; * Speciality\_id (N, О) – Специальность; * EducationInstitution\_id (N, Н) – Учебное заведение. |
| **Пример** |  |

* + 1. Создание данных о сертификатах сотрудника

**POST api/Certificate**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N,Н) – Ид человека; * MedWorker\_id (N, Н) – Ид сотрудника. * CertificateReceipDate (D, О) – Дата получения сертификата; * CertificateNumber (T, О) – Номер сертификата; * CertificateSeries (T, Н) – Серия сертификата; * OtherEducationalInstitution (T, Н) – Иное учебное заведение; * Speciality\_id (N, О) – Специальность; * EducationInstitution\_id (N, Н) – Учебное заведение |
| **Выполняются действия** | Производятся проверки:   * Должен быть указан хотя бы один из параметров Person\_id или MedWorker\_id должен быть заполнен. Иначе ошибка. * Если не заполнены обязательные параметры, то ошибка. * Если не найден указанный человек или медсотрудник, то ошибка.   Если есть ошибки, то возвращается код ошибки.  Иначе: создаются данные о курсах переподготовки сотрудника. |
| **Ответ** | Если были ошибки, то коды ошибок.  Иначе: id (N, О) – Ид записи о сертификате; |
| **Пример** |  |

* + 1. Изменение данных о сертификатах сотрудника

**PUT api/Certificate**

|  |  |
| --- | --- |
| **Входящие параметры** | * id (N, О) – Ид записи о сертификате; * CertificateReceipDate (D, О) – Дата получения сертификата; * CertificateNumber (T, О) – Номер сертификата; * CertificateSeries (T, Н) – Серия сертификата; * OtherEducationalInstitution (T, Н) – Иное учебное заведение; * Speciality\_id (N, О) – Специальность; * EducationInstitution\_id (N, Н) – Учебное заведение |
| **Выполняются действия** | Производятся проверки:   * Если не заполнены обязательные параметры, то ошибка. * Если не найден идентификатор сертификата, то ошибка.   Если есть ошибки, то данные не изменяются.  Иначе: изменяются данные о сертификате сотрудника. |
| **Ответ** | Если были ошибки, то коды ошибок.  Иначе: код 0. |
| **Пример** |  |

* + 1. Удаление данных о сертификатах сотрудника

**DELETE api/Certificate**

|  |  |
| --- | --- |
| **Входящие параметры** | * id (N, О) – Ид записи о сертификате; |
| **Выполняются действия** | Производится проверка: если указанный идентификатор сертификата не найден, то ошибка. Данные не удаляются.  Иначе: Указанная запись удаляется. |
| **Ответ** | Если при выполнении метода были ошибки, то возвращается код ошибки.  Если данные удалены, то возвращается код 0. |

* + 1. Получение данных о сертификатах сотрудника

**GET api/Certificate**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N,Н) – Ид человека; * MedWorker\_id (N, Н) – Ид сотрудника. |
| **Ответ** | Производится проверка: хотя бы один из параметров Person\_id или MedWorker\_id должен быть заполнен. Иначе ошибка.  В ответе возвращается записи со следующими параметрами:   * id (N, О) – Ид записи о сертификате; * MedWorker\_id (N, О) – Ид сотрудника; * Person\_id (N, О) – Ид человека; * CertificateReceipDate (D, О) – Дата получения сертификата; * CertificateNumber (T, О) – Номер сертификата; * CertificateSeries (T, Н) – Серия сертификата; * OtherEducationalInstitution (T, Н) – Иное учебное заведение; * Speciality\_id (N, О) – Специальность; * EducationInstitution\_id (N, Н) – Учебное заведение. |
| **Пример** |  |

* + 1. Создание данных о квалификационных категориях сотрудника

**POST api/QualificationCategory**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N,Н) – Ид человека; * MedWorker\_id (N, Н) – Ид сотрудника. * Category\_id (N, О) – Квалификационная категория; * Speciality\_id (N, О) – Специальность; * AssigmentDate (D, О) – Дата присвоения. |
| **Выполняются действия** | Производятся проверки:   * Должен быть указан хотя бы один из параметров Person\_id или MedWorker\_id должен быть заполнен. Иначе ошибка. * Если не заполнены обязательные параметры, то ошибка. * Выполняется проверка полученных входящих параметров, значения которых заданы идентификаторами. Если в соответствующих объектах РМИС эти значения не определены, то ошибка.   Если есть ошибки, то создание данных не выполняется, возвращается код ошибки.  Иначе: создаются данные о квалификационных категориях сотрудника. |
| **Ответ** | В ответе возвращается:  Если были ошибки, то коды ошибок.  Иначе: id (N, О) – Ид записи о квалификационной категории; |
| **Пример** |  |

* + 1. Изменение данных о квалификационных категориях (сертификатах) сотрудника

**PUT api/QualificationCategory**

|  |  |
| --- | --- |
| **Входящие параметры** | * id (N, О) – Ид записи о квалификационной категории; * Category\_id (N, О) – Квалификационная категория; * Speciality\_id (N, О) – Специальность; * AssigmentDate (D, О) – Дата присвоения. |
| **Выполняются действия** | Производятся проверки:   * Если не заполнены обязательные параметры, то ошибка. * Выполняется проверка полученных входящих параметров, значения которых заданы идентификаторами. Если в соответствующих объектах РМИС эти значения не определены, то ошибка.   Если есть ошибки, то изменение данных не выполняется.  Иначе: изменяются данные о квалификационных категориях сотрудника. |
| **Ответ** | Если были ошибки, то коды ошибок.  Иначе: код 0. |
| **Пример** |  |

* + 1. Удаление данных о квалификационных категориях сотрудника

**DELETE api/QualificationCategory**

|  |  |
| --- | --- |
| **Входящие параметры** | * id (N, О) – Ид записи о квалификационной категории; |
| **Выполняются действия** | Производится проверка: если указанный идентификатор квалификационной категории не найден, то ошибка. Данные не удаляются.  Иначе: Указанная запись удаляется. |
| **Ответ** | Если при выполнении метода были ошибки, то возвращается код ошибки.  Если данные удалены, то возвращается код 0. |

* + 1. Получение данных о квалификационных категориях сотрудника

**GET api/QualificationCategory**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N,Н) – Ид человека; * MedWorker\_id (N, Н) – Ид сотрудника. |
| **Ответ** | Производится проверка: хотя бы один из параметров Person\_id или MedWorker\_id должен быть заполнен. Иначе ошибка.  В ответе возвращается записи со следующими параметрами:   * id (N, О) – Ид записи о квалификационной категории; * MedWorker\_id (N, О) – Ид сотрудника; * Person\_id (N, О) – Ид человека; * Category\_id (N, О) – Квалификационная категория; * Speciality\_id (N, О) – Специальность; * AssigmentDate (D, О) – Дата присвоения. |
| **Пример** |  |

* + 1. Создание данных о наградах сотрудника

**POST api/Reward**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N,Н) – Ид человека; * MedWorker\_id (N, Н) – Ид сотрудника. * date (D, О) – Дата; * name (T, О) – Название награды; * number (T, О) – Номер; * FRMPNomination\_id (N, О) – Номинация. |
| **Выполняются действия** | Производятся проверки:   * Должен быть указан хотя бы один из параметров Person\_id или MedWorker\_id должен быть заполнен. Иначе ошибка. * Если не заполнены обязательные параметры, то ошибка. * Если не найден указанный человек или медсотрудник, то ошибка.   Если есть ошибки, то возвращается код ошибки.  Иначе: создаются данные о наградах сотрудника. |
| **Ответ** | Если были ошибки, то возвращаются коды ошибок.  Иначе: id (N, О) – Ид записи о награде |
| **Пример** |  |

* + 1. Изменение данных о наградах сотрудника

**PUT api/ Reward**

|  |  |
| --- | --- |
| **Входящие параметры** | * id (N, О) – Ид записи о награде * date (D, О) – Дата; * name (T, О) – Название награды; * number (T, О) – Номер; * FRMPNomination\_id (N, О) – Номинация. |
| **Выполняются действия** | Производятся проверки:   * Если не заполнены обязательные параметры, то ошибка. * Выполняется проверка полученных входящих параметров, значения которых заданы идентификаторами .Если в соответствующих объектах РМИС указанного значения нет, то ошибка.   Если есть ошибки, то данные не изменяются.  Иначе: изменяются данные о наградах сотрудника. |
| **Ответ** | Если были ошибки, то возвращаются коды ошибок.  Иначе: код 0. |
| **Пример** |  |

* + 1. Удаление данных о наградах сотрудника

**DELETE api/ Reward**

|  |  |
| --- | --- |
| **Входящие параметры** | * id (N, О) – Ид записи о награде; |
| **Выполняются действия** | Производится проверка: если указанный идентификатор награды не найден, то ошибка. Данные не удаляются.  Иначе: данные запись удаляется. |
| **Ответ** | Если при выполнении метода были ошибки, то возвращается код ошибки.  Если данные удалены, то возвращается код 0. |

* + 1. Получение данных о наградах сотрудника

**GET api/Reward**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N,Н) – Ид человека; * MedWorker\_id (N, Н) – Ид сотрудника. |
| **Ответ** | Производится проверка: хотя бы один из параметров Person\_id или MedWorker\_id должен быть заполнен. Иначе ошибка.  В ответе возвращается записи со следующими параметрами:   * id (N, О) – Ид записи о награде; * MedWorker\_id (N, О) – Ид сотрудника; * Person\_id (N, О) – Ид человека; * date (D, О) – Дата; * name (T, О) – Название награды; * number (T, О) – Номер; * FRMPNomination\_id (N, О) – Номинация. |
| **Пример** |  |

* + 1. Создание данных об аккредитации сотрудника

**POST api/ Accreditation**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N,Н) – Ид человека; * MedWorker\_id (N, Н) – Ид сотрудника. * DocumentNumber (T, О) – Номер документа (текстовое); * DocumentSeries (T, О) – Серия документа (текстовое); * RegNumber (T, Н) – Регистрационный номер (текстовое); * AccreditationType\_id (N, О) – Вид аккредитации; * DiplomaSpeciality\_id (N, О) – Специальность; * ProfStandard\_id (N, О) – Профессиональный стандарт; * EducationInstitution\_id (N, О) – Учебное заведение; * OtherEducationalInstitution (T, Н)– Иное учебное заведение; * PassDate (D, О) - Дата проведения.. |
| **Выполняются действия** | Производятся проверки:   * Должен быть указан хотя бы один из параметров Person\_id или MedWorker\_id должен быть заполнен. Иначе ошибка. * Если не заполнены обязательные параметры, то ошибка. * Выполняется проверка полученных входящих параметров, значения которых заданы идентификаторами .Если в соответствующих объектах РМИС указанного значения нет, то ошибка.   Если есть ошибки, то данные не создаются, возвращается код ошибки.  Иначе: создаются данные об аккредитации сотрудника. |
| **Ответ** | Если были ошибки, то возвращаются коды ошибок.  Иначе: id (N, О) – Ид записи об аккредитации |
| **Пример** |  |

* + 1. Изменение данных об аккредитации сотрудника

**PUT api/Accreditation**

|  |  |
| --- | --- |
| **Входящие параметры** | * id (N, О) – Ид записи о награде * DocumentNumber (T, О) – Номер документа (текстовое); * DocumentSeries (T, О) – Серия документа (текстовое); * RegNumber (T, Н) – Регистрационный номер (текстовое); * AccreditationType\_id (N, О) – Вид аккредитации; * DiplomaSpeciality\_id (N, О) – Специальность; * ProfStandard\_id (N, О) – Профессиональный стандарт; * EducationInstitution\_id (N, О) – Учебное заведение; * OtherEducationalInstitution (T, Н)– Иное учебное заведение; * PassDate (D, О) - Дата проведения. |
| **Выполняются действия** | Производятся проверки:   * Если не заполнены обязательные параметры, то ошибка. * Выполняется проверка полученных входящих параметров, значения которых заданы идентификаторами .Если в соответствующих объектах РМИС указанного значения нет, то ошибка.   Если есть ошибки, то данные не изменяются, возвращается код ошибки.  Иначе: изменяются данные об аккредитации сотрудника. |
| **Ответ** | Если были ошибки, то возвращается коды ошибок.  Иначе: код 0. |
| **Пример** |  |

* + 1. Удаление данных об аккредитации сотрудника

**DELETE api/ Accreditation**

|  |  |
| --- | --- |
| **Входящие параметры** | * id (N, О) – Ид записи об аккредитации; |
| **Выполняются действия** | Производится проверка: если указанный идентификатор аккредитации не найден, то ошибка. Данные не удаляются.  Иначе: Указанная запись удаляется. |
| **Ответ** | Если при выполнении метода были ошибки, то возвращается код ошибки.  Если данные удалены, то возвращается код 0. |

* + 1. Получение данных об аккредитации сотрудника

**GET api/Accreditation**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N,Н) – Ид человека; * MedWorker\_id (N, Н) – Ид сотрудника. |
| **Ответ** | Производится проверка: хотя бы один из параметров Person\_id или MedWorker\_id должен быть заполнен. Иначе ошибка.  В ответе возвращается записи со следующими параметрами:   * id (N, О) – Ид записи об аккредитации; * MedWorker\_id (N, О) – Ид сотрудника; * Person\_id (N, О) – Ид человека; * DocumentNumber (T, О) – Номер документа (текстовое); * DocumentSeries (T, О) – Серия документа (текстовое); * RegNumber (T, Н) – Регистрационный номер (текстовое); * AccreditationType\_id (N, О) – Вид аккредитации; * DiplomaSpeciality\_id (N, О) – Специальность; * ProfStandard\_id (N, О) – Профессиональный стандарт; * EducationInstitution\_id (N, О) – Учебное заведение; * OtherEducationalInstitution (T, Н)– Иное учебное заведение; * PassDate (D, О) - Дата проведения. |
| **Пример** |  |

* 1. Описание методов работы с должностью
  2. 1. Создание должности

**POST api/Post**

Создание должности

|  |  |
| --- | --- |
| **Входящие параметры** | * Сode (S, О) – код * Name (S, О) - наименование * FRMPPost\_id (N, H) – должность ФРМП (значение из справочника persis.frmppost) * PostKind\_id (N, О) – вид должности (значение из справочника persis.postkind) * ProfessionalGroup\_id (N, H) – профессиональная группа (значение из справочника persis.ProfessionalGroup) * TarifList (N, H) – тарификационный лист * Speciality\_id (N, H) – специальность (значение из справочника persis.Speciality) * PrimaryHealthCare (N, O) – Флаг Первичное звено. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * MaxPopulation (N, H) – максимальная численность прикрепления * Post\_Nick (Т, О) – Наименование должности для ЛВН <http://redmine.swan.perm.ru/issues/136823> * begDT (D, О) – дата открытия |
| **Ответ** | **Успешный ответ**  Возвращает идентификатор должности   * Post\_id (N, О) – идентификатор должности |

* + 1. Изменение должности

**PUT api/Post**

Изменение должности

|  |  |
| --- | --- |
| **Входящие параметры** | * Post\_id (N, О) – идентификатор должности * Сode (S, H) – код * Name (S, Н) - наименование * FRMPPost\_id (N, H) – должность ФРМП (значение из справочника persis.frmppost) * PostKind\_id (N, H) – вид должности (значение из справочника persis.postkind) * ProfessionalGroup\_id (N, H) – профессиональная группа (значение из справочника persis.ProfessionalGroup) * TarifList (N, H) – тарификационный лист * Speciality\_id (N, H) – специальность (значение из справочника persis.Speciality) * PrimaryHealthCare (N, H) – Флаг Первичное звено * MaxPopulation (N, H) – максимальная численность прикрепления * Post\_Nick (Т, О) – Наименование должности для ЛВН <http://redmine.swan.perm.ru/issues/136823> * begDT (D, О) – дата открытия |
| **Ответ** | **Успешный ответ**  Возвращается пустой ответ с кодом ошибки 0 |

* + 1. Получение должности по коду

**GET api/PostByCode**

Получение должности по коду

|  |  |
| --- | --- |
| **Входящие параметры** | * Сode (S, О) – код |
| **Ответ** | **Успешный ответ**  Данные должности   * Сode (S, О) – код * Name (S, Н) - наименование * FRMPPost\_id (N, H) – должность ФРМП (значение из справочника persis.frmppost) * PostKind\_id (N, О) – вид должности (значение из справочника persis.postkind) * ProfessionalGroup\_id (N, H) – профессиональная группа (значение из справочника persis.ProfessionalGroup) * TarifList (N, H) – тарификационный лист * Speciality\_id (N, H) – специальность (значение из справочника persis.Speciality) * PrimaryHealthCare (N, O) – Флаг Первичное звено * MaxPopulation (N, H) – максимальная численность прикрепления |

* + 1. Получение должности по идентификатору

**GET api/PostByid**

Получение должности по коду

|  |  |
| --- | --- |
| **Входящие параметры** | * Post\_id (N, О) – идентификатор должности |
| **Ответ** | **Успешный ответ**  Данные должности   * Сode (S, О) – код * Name (S, Н) - наименование * FRMPPost\_id (N, H) – должность ФРМП (значение из справочника persis.frmppost) * PostKind\_id (N, О) – вид должности (значение из справочника persis.postkind) * ProfessionalGroup\_id (N, H) – профессиональная группа (значение из справочника persis.ProfessionalGroup) * TarifList (N, H) – тарификационный лист * Speciality\_id (N, H) – специальность (значение из справочника persis.Speciality) * PrimaryHealthCare (N, O) – Флаг Первичное звено * MaxPopulation (N, H) – максимальная численность прикрепления |

* 1. Описание методов работы со штатным расписанием
  2. 1. Создание строки штатного расписания

**POST api/Staff**

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuSection\_id (N, О) - Идентификатор отделения * Post\_id (N,O) – Идентификатор должности * isDummyStaff (N, O) – Флаг фиктивной ставки. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * PayType\_id (D, H) – Идентификатор источника финансирования * Rate (N, O) – Количество ставок * FRMPSubdivision\_id (N, H) – Идентификатор типа подразделения. Обязательно для заполнения, если количество ставок >= 0,2 * BeginDate (D, O) – Дата создания * EndDate (D, H) – Дата закрытия * MedicalCareKind\_id (N, H) – Идентификатор вида медицинской помощи. Обязательно для заполнения, если должность врачебная. * Сomments (T, H) - Комментарий * RateNorm (N, H) – Количество ставок по нормативу * LeadershipBonusPercent (N, H) - Процент надбавки за руководство * SalaryReductionPercent (N, H) - Процент уменьшения должностного оклада * IsVillageBonus (N, O) – Флаг «Надбавка за работу на селе». Возможные значения: 0 и 1, - где 0 – нет, 1 – да * SpecialLabourType (N, H) - Учитываемый специальный тип стажа * VacancyOfficialSalary (N, H) - Оклад у вакантных должностей * RateFinancing (N, H) - Финансирование ставки * UETCount (N, H) - Количество УЕТ |
| **Ответ** | **Успешный ответ**  Возвращает идентификатор строки штатного расписания   * Staff\_id (N, О) – Идентификатор строки штатного расписания |

* + 1. Изменение строки штатного расписания

**PUT api/Staff**

|  |  |
| --- | --- |
| **Входящие параметры** | * Staff\_id (N, О) – Идентификатор строки штатного расписания * isDummyStaff (N, H) – Флаг фиктивной ставки * PayType\_id (D, H) – Идентификатор источника финансирования * Rate (N, H) – Количество ставок * FRMPSubdivision\_id (N, H) – Идентификатор типа подразделения. Обязательно для заполнения, если количество ставок >= 0,2 * BeginDate (D, H) – Дата создания * EndDate (D, H) – Дата закрытия * MedicalCareKind\_id (N, H) – Идентификатор вида медицинской помощи. Обязательно для заполнения, если должность врачебная. * Сomments (T, H) - Комментарий * RateNorm (N, H) – Количество ставок по нормативу * LeadershipBonusPercent (N, H) - Процент надбавки за руководство * SalaryReductionPercent (N, H) - Процент уменьшения должностного оклада * IsVillageBonus (N, H) – Флаг «Надбавка за работу на селе» * SpecialLabourType (N, H) - Учитываемый специальный тип стажа * VacancyOfficialSalary (N, H) - Оклад у вакантных должностей * RateFinancing (N, H) - Финансирование ставки * UETCount (N, H) - Количество УЕТ |
| **Ответ** | **Успешный ответ**  Возвращается пустой ответ с кодом ошибки 0 |

* + 1. Получение строки штатного расписания по идентификатору

**GET api/StaffByid**

|  |  |
| --- | --- |
| **Входящие параметры** | * Staff\_id (N, О) - Идентификатор строки штатного расписания |
| **Ответ** | **Успешный ответ**  Возвращает список строк штатного расписания   * Staff\_id (N, О) – Идентификатор строки штатного расписания * isDummyStaff (N, O) – Флаг фиктивной ставки * PayType\_id (D, H) – Идентификатор источника финансирования * Rate (N, O) – Количество ставок * FRMPSubdivision\_id (N, H) – Идентификатор типа подразделения. Обязательно для заполнения, если количество ставок >= 0,2 * BeginDate (D, O) – Дата создания * EndDate (D, H) – Дата закрытия * MedicalCareKind\_id (N, H) – Идентификатор вида медицинской помощи. Обязательно для заполнения, если должность врачебная. * Сomments (T, H) - Комментарий * RateNorm (N, H) – Количество ставок по нормативу * LeadershipBonusPercent (N, H) - Процент надбавки за руководство * SalaryReductionPercent (N, H) - Процент уменьшения должностного оклада * IsVillageBonus (N, O) – Флаг «Надбавка за работу на селе» * SpecialLabourType (N, H) - Учитываемый специальный тип стажа * VacancyOfficialSalary (N, H) - Оклад у вакантных должностей * RateFinancing (N, H) - Финансирование ставки * UETCount (N, H) - Количество УЕТ |

* + 1. Получение строки штатного расписания по месту работы

**GET api/StaffByMedStaffFact**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedStaffFact\_id (N, О) – Место работы врача |
| **Ответ** | **Успешный ответ**  Возвращает строку штатного расписания   * Staff\_id (N, О) – Идентификатор строки штатного расписания |

* + 1. Получение строки штатного расписания по отделению МО

**GET api/StaffByLpuSection**

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuSection\_id (N, О) - Идентификатор отделения |
| **Ответ** | **Успешный ответ**  Возвращает список строк штатного расписания   * Staff\_id (N, О) – Идентификатор строки штатного расписания |

* + 1. Создание выплаты для строки штатного расписания

**POST api/Payment**

|  |  |
| --- | --- |
| **Входящие параметры** | * Staff\_id (N, О) – Идентификатор строки штатного расписания * PaymentKind\_id (N,O) – Вид выплаты * PaymentPercent (N, O) – Процент |
| **Ответ** | **Успешный ответ**  Возвращает идентификатор выплаты   * Payment\_id (N, О) – Идентификатор выплаты. Для связи со строкой штатного расписания происходит запись в таблицу persis.StaffPayment |

* + 1. Изменение выплаты

**PUT api/Payment**

|  |  |
| --- | --- |
| **Входящие параметры** | * Payment\_id (N, О) – Идентификатор выплаты * PaymentKind\_id (N, H) – Вид выплаты * PaymentPercent (N, H) – Процент |
| **Ответ** | **Успешный ответ**  Возвращается пустой ответ с кодом ошибки 0 |

* + 1. Получение выплат по строке штатного расписания

**GET api/PaymentByStaff**

|  |  |
| --- | --- |
| **Входящие параметры** | * Staff\_id (N, О) – Идентификатор строки штатного расписания |
| **Ответ** | **Успешный ответ**  Список выплат:   * Payment\_id (N, О) – Идентификатор выплаты. Связь со строкой штатного расписания берем из таблицы persis.StaffPayment |

* 1. Описание методов работы с местом работы
  2. 1. Создание места работы

**POST api/MedStaffFact**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedPesonal\_id (N, О) – идентификатор сотрудника * Staff\_id (N, O) – идентификатор строки штатного расписания * TabCode(S, H) – табельный номер * IsDummyWP (N, ОO) – флаг фиктивного места работы. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * PostOccupationType \_id(N, O) – тип занятия должности (ссылка на справочник persis.PostOccupationType) * Rate(N, O) – ставка * WorkMode\_id(N, H) – режим работы (ссылка на справочник persis.workmode) * MilitaryRelation\_id(N, H) – отношение к военной службе (ссылка на справочник persis.MilitaryRelation) * FRMPSubdivision\_id(N, H) – тип подразделения (ссылка на справочник persis.FRMPSubdivision) * MedSpecOms\_id(N, H) – специальность врача * Population(N, H) – численность прикрепления * IsOMS(N, О) – работает в ОМС, значения Да/Нет. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * DisableWorkPlaceChooseInDocuments(N, О) – флаг запрета выбора места работы в документах. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * Comments(S, H) – примечание * RecType\_id (N, H) – тип записи(ссылка на справочник RecType) * PriemTime(N, H) – время приема * IsDirRec(N, O) – разрешать запись к врачу через направление. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * IsQueueOnFree(N, O) – флаг, позволение посещения в очередь при наличии свободных бирок. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * IsNotReception(N, О) – флаг, не вести участковый прием. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * Descr(S, H) – примечание врача * Contacts(S, H) – контактная информация * DLOBeginDate(D, H) – дата включения в ДЛО * DLOEndDate(D, H) – дата исключения из ДЛО * CommonLabourYears(N, H) – непрерывный медицинский стаж на момент начала работы (ГГ) * CommonLabourMonths(N, H) – непрерывный медицинский стаж на момент начала работы (ММ) * CommonLabourDays(N, H) – непрерывный медицинский стаж на момент начала работы (ДД) * SpecialLabourDays(N, H) – в том числе специальный стаж на момент начала работы (ГГ) * SpecialLabourMonths(N, H) – в том числе специальный стаж на момент начала работы (ММ) * SpecialLabourYears(N, H) – в том числе специальный стаж на момент начала работы (ДД) * ArriveRecordType\_id(N, O) – Запись на начало(ссылка на справочник persis.ArriveRecordType) * ArriveOrderNumber(S, O) – Номер приказа на начало * BeginDate(D, O) – Дата начала * AdditionalAgreementDate(D, H) – Дата заключения доп. соглашения * AdditionalAgreementNumber(N. H) – Номер доп. соглашения * LeaveRecordType\_id(N, H ) – Запись на окончание(ссылка на справочник persis.LeaveRecordType) * LeaveOrderNumber(S, H) – Номер приказа на окончание * EndDate(D, H) – Дата окончания * OfficialSalary(N, H ) – Должностной оклад * QualificationLevel(N, H ) – Квалификационный уровень * MedStaffFactOuter\_id(N, О) – идентификатор места работы во внешней МИС |
| **Ответ** | **Успешный ответ**  Возвращает идентификатор места работы   * MedStaffFact\_id (N, О) – идентификатор места работы |

* + 1. Изменение места работы

**PUT api/MedStaffFact**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedStaffFact\_id (N, О) – идентификатор места работы * MedPesonal\_id (N, H) – идентификатор сотрудника * Staff\_id (N, H) – идентификатор строки штатного расписания * TabCode(S, H) – табельный номер * IsDummyWP (N, H) – флаг фиктивного места работы * PostOccupationType \_id(N, H) – тип занятия должности (ссылка на справочник persis.PostOccupationType) * Rate(N, H) – ставка * WorkMode\_id(N, H) – режим работы (ссылка на справочник persis.workmode) * MilitaryRelation\_id(N, H) – отношение к военной службе (ссылка на справочник persis.MilitaryRelation) * FRMPSubdivision\_id(N, H) – тип подразделения (ссылка на справочник persis.FRMPSubdivision) * MedSpecOms\_id(N, H) – специальность врача * Population(N, H) – численность прикрепления * IsOMS(N, H) – работает в ОМС, значения Да/Нет * DisableWorkPlaceChooseInDocuments(N, H) – флаг запрета выбора места работы в документах * Comments(S, H) – примечание * RecType\_id (N, H) – тип записи(ссылка на справочник RecType) * PriemTime(N, H) – время приема * IsDirRec(N, H) – разрешать запись к врачу через направление * IsQueueOnFree(N, H) – флаг, позволение посещения в очередь при наличии свободных бирок * IsNotReception(N, H) – флаг, не вести участковый прием * Descr(S, H) – примечание врача * Contacts(S, H) – контактная информация * DLOBeginDate(D, H) – дата включения в ДЛО * DLOEndDate(D, H) – дата исключения из ДЛО * CommonLabourYears(N, H) – непрерывный медицинский стаж на момент начала работы (ГГ) * CommonLabourMonths(N, H) – непрерывный медицинский стаж на момент начала работы (ММ) * CommonLabourDays(N, H) – непрерывный медицинский стаж на момент начала работы (ДД) * SpecialLabourDays(N, H) – в том числе специальный стаж на момент начала работы (ГГ) * SpecialLabourMonths(N, H) – в том числе специальный стаж на момент начала работы (ММ) * SpecialLabourYears(N, H) – в том числе специальный стаж на момент начала работы (ДД) * ArriveRecordType\_id(N, H) – Запись на начало(ссылка на справочник persis.ArriveRecordType) * ArriveOrderNumber(S, H) – Номер приказа на начало * BeginDate(D, H) – Дата начала * AdditionalAgreementDate(D, H) – Дата заключения доп. соглашения * AdditionalAgreementNumber(N. H) – Номер доп. соглашения * LeaveRecordType\_id(N, H ) – Запись на окончание(ссылка на справочник persis.LeaveRecordType) * LeaveOrderNumber(S, H) – Номер приказа на окончание * EndDate(D, H) – Дата окончания * OfficialSalary(N, H ) – Должностной оклад * QualificationLevel(N, H ) – Квалификационный уровень * MedStaffFactOuter\_id(N, H) – идентификатор места работы во внешней МИС |
| **Ответ** | **Успешный ответ**   * Возвращается пустой ответ с кодом ошибки 0 |

* + 1. Получение места работы по идентификатору

**GET api/MedStaffFactByid**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedStaffFact\_id (N, Н) – Идентификатор места работы врача в РИШ; * MedStaffFactOuter\_id (N, Н) – Идентификатор места работы врача в сторонней МИС. * Person\_id (N, Н) – Ид человека. |
| **Ответ** | Если не передан ни один параметр, то ошибка.  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список мест работы врачей с информацией:  Возвращает данные места работы   * MedStaffFact\_id (N, О) – Идентификатор места работы врача в РИШ; * MedPesonal\_id (N, О) – идентификатор сотрудника * Staff\_id (N, O) – идентификатор строки штатного расписания * TabCode(S, H) – табельный номер * IsDummyWP (N, H) – флаг фиктивного места работы * PostOccupationType \_id(N, O) – тип занятия должности (ссылка на справочник persis.PostOccupationType) * Rate(N, O) – ставка * WorkMode\_id(N, H) – режим работы (ссылка на справочник persis.workmode) * MilitaryRelation\_id(N, H) – отношение к военной службе (ссылка на справочник persis.MilitaryRelation) * FRMPSubdivision\_id(N, H) – тип подразделения (ссылка на справочник persis.FRMPSubdivision) * MedSpecOms\_id(N, H) – специальность врача * Population(N, H) – численность прикрепления * IsOMS(N, H) – работает в ОМС, значения Да/Нет * DisableWorkPlaceChooseInDocuments(N, H) – флаг запрета выбора места работы в документах * Comments(S, H) – примечание * RecType\_id (N, H) – тип записи(ссылка на справочник RecType) * PriemTime(N, H) – время приема * IsDirRec(N, O) – разрешать запись к врачу через направление * IsQueueOnFree(N, O) – флаг, позволение посещения в очередь при наличии свободных бирок * IsNotReception(N, H) – флаг, не вести участковый прием * Descr(S, H) – примечание врача * Contacts(S, H) – контактная информация * DLOBeginDate(D, H) – дата включения в ДЛО * DLOEndDate(D, H) – дата исключения из ДЛО * CommonLabourYears(N, H) – непрерывный медицинский стаж на момент начала работы (ГГ) * CommonLabourMonths(N, H) – непрерывный медицинский стаж на момент начала работы (ММ) * CommonLabourDays(N, H) – непрерывный медицинский стаж на момент начала работы (ДД) * SpecialLabourDays(N, H) – в том числе специальный стаж на момент начала работы (ГГ) * SpecialLabourMonths(N, H) – в том числе специальный стаж на момент начала работы (ММ) * SpecialLabourYears(N, H) – в том числе специальный стаж на момент начала работы (ДД) * ArriveRecordType\_id(N, O) – Запись на начало(ссылка на справочник persis.ArriveRecordType) * ArriveOrderNumber(S, O) – Номер приказа на начало * BeginDate(D, O) – Дата начала * AdditionalAgreementDate(D, H) – Дата заключения доп. соглашения * AdditionalAgreementNumber(N. H) – Номер доп. соглашения * LeaveRecordType\_id(N, H ) – Запись на окончание(ссылка на справочник persis.LeaveRecordType) * LeaveOrderNumber(S, H) – Номер приказа на окончание * EndDate(D, H) – Дата окончания * OfficialSalary(N, H ) – Должностной оклад * QualificationLevel(N, H ) – Квалификационный уровень * MedStaffFactOuter\_id(N, О) – идентификатор места работы во внешней МИС; * LpuSection\_id (Т, О) – Идентификатор отделения МО места работы врача. |

* + 1. Получение списка мест работы по МО и профилю

**GET api/MedStaffFactListByMOandProfile**

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N, O) – Идентификатор МО * LpuSectionProfile (N, O) – Идентификатор профиля отделения |
| **Ответ** | **Успешный ответ**  Возвращает список мест работы, штатное расписание которых относится к отделениям с искомым профилем.   * MedStaffFact\_id (N, О) – идентификатор места работы |

* + 1. Создание невыплаты

**POST api/SkipPayment**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedStaffFact\_id (N, О) – идентификатор места работы * SkipPaymentReason\_id(N, О) – причина невыплат (сслыка на справочник persis.SkipPaymentReason) * StartDate (D, О) – дата начала * EndDate(D, О) – дата окончания |
| **Ответ** | **Успешный ответ**  Возвращает идентификатор невыплаты   * SkipPayment\_id (N, О) – идентификатор невыплаты |

* + 1. Изменение невыплаты

**PUT api/SkipPayment**

|  |  |
| --- | --- |
| **Входящие параметры** | * SkipPayment\_id (N, О) – идентификатор невыплаты * MedStaffFact\_id (N, H) – идентификатор места работы * SkipPaymentReason\_id(N, H) – причина невыплат (сслыка на справочник persis.SkipPaymentReason) * StartDate (D, H) – дата начала * EndDate(D, H) – дата окончания |
| **Ответ** | **Успешный ответ**   * Возвращается пустой ответ с кодом ошибки 0 |

* + 1. Получение невыплаты по идентификатору

**GET api/SkipPaymentByid**

|  |  |
| --- | --- |
| **Входящие параметры** | * SkipPayment\_id (N, О) – идентификатор невыплаты |
| **Ответ** | **Успешный ответ**  Возвращает данные невыплаты   * MedStaffFact\_id (N, О) – идентификатор места работы * SkipPaymentReason\_id(N, О) – причина невыплат (сслыка на справочник persis.SkipPaymentReason) * StartDate (D, О) – дата начала * EndDate(D, О) – дата окончания0 |

* + 1. Получение навыплат по месту работы

**GET api/SkipPaymentByMedStaffFact**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedStaffFact\_id (N, О) – идентификатор места работы |
| **Ответ** | **Успешный ответ**  Возвращает список невыплат   * SkipPayment\_id(N, О) – идентификатор невыплаты |

* + 1. Создание выплаты для места работы

**POST api/PaymentMedStaffFact**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedStaffFact\_id (N, О) – Идентификатор места работы * PaymentKind\_id (N,O) – Вид выплаты * PaymentPercent (N, O) – Процент |
| **Ответ** | **Успешный ответ**  Возвращает идентификатор выплаты   * Payment\_id (N, О) – Идентификатор выплаты. Для связи с местом работы происходит запись в таблицу persis.WorkPlacePayment |

* + 1. Получение выплат по месту работы

**GET api/PaymentByMedStaffFact**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedStaffFact\_id (N, О) – Идентификатор места работы |
| **Ответ** | **Успешный ответ**  Список выплат:   * Payment\_id (N, О) – Идентификатор выплаты. Связь со строкой штатного расписания берем из таблицы persis.WorkPlacePayment |

* 1. Описание методов работы с паспортом МО
  2. 1. Изменение паспорта МО

**PUT api/PassportMO/PassportMO**

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. * Lpu\_f003mcod (N, H) – Федеральный реестровый код * Lpu\_begDate (D, H) – Дата начала деятельности. * Oktmo\_id (N, H) - Идентификатор ОКТМО. Значение из справочника dbo.Oktmo * Org\_OKPO (T, H) – Идентификатор ОКПО * Org\_OKATO (T, H) – Идентификатор ОКАТО * Okogu\_id (N, H) – Идентификатор ОКОГУ. Значение справочника dbo.Okogu * Okved\_id (N, H) – Идентификатор ОКВЭД. Значение справочника dbo.Okved * Okfs\_id (N, H) – Идентификатор ОКФС. Значение справочника dbo.Okfs * Org\_OKDP (T, H) – Идентификатор ОКДП * Okopf\_id (N, H) – Идентификатор ОКОПФ. Значение справочника dbo.Okopf * Lpu\_PensRegNum (N, H) - Рег номер в ПФ РФ * UAddress\_id (N, H) – Юридический адрес * PAddress\_id (N, H) – Фактический адрес * Org\_INN (T, H) - ИНН * Org\_Email (T, H) – Адрес электронной почты * Org\_Www (T, H) – Адрес сайта * DepartAffilType\_id (N, H) – Ведомственная принадлежность. Значение из справочника fed.DepartAffilType * Org\_KPP (T, H) - КПП * Lpu\_RegDate (D, H) – Дата регистрации * Lpu\_DocReg (T, H) – Документ о регистрации * Org\_OGRN (T, H) – Идентификатор ОГРН * Org\_id (N, H) – Регистрирующий орган (значение справочника dbo.Org) * Org\_Phone (T, H) -Телефон * LpuPmuType\_id (N, H) – Тип МО для ПМУ. Значение справочника fed.LpuPmuType * MesAgeLpuType\_id (N, H) – Тип МО по возрасту. Значение справочника dbo.MesAgeLpuType * LpuSubjectionLevel\_id (N, Н) – Идентификатор уровня подчиненности МО. Значение из справочника dbo.LpuSubjectionLevel * Lpu\_gid (N, H) – идентификатор головного учеждения. * Lpu\_VizitFact (N, H) – количество посещений в смену. * InstitutionLevel\_id (N, H) – идентификатор уровня учреждения в иерархии сети. Значение из справочника passport.InstitutionLevel * PasportMO\_MaxDistansePoint (N, H) - Расстояние до наиболее удаленной точки территориального обслуживания (км) * DLocationLpu\_id (N, H) – Идентификатор местоположения учреждения. Значение из справочника fed.DLocationLpu * PasportMO\_IsVideo (N, H) – Наличие видеонаблюдения территорий и помещений. Значение типа флаг. * PasportMO\_IsMetalDoors (N, H) – Наличие металлических входных дверей. Значение типа флаг. * PasportMO\_IsSecur (N, H) – Наличие охраны. Значение типа флаг. * PasportMO\_IsAccompanying (N, H) – Проживание сопровождающих лиц. Значение типа флаг. * PasportMO\_IsFenceTer (N, H) – Ограждение территории. Значение типа флаг. * PasportMO\_IsTerLimited(N, H) – Приспособленность территории для пациентов с ограниченными возможностями. Значение типа флаг. |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0 |

* + 1. Добавление периода по ОМС

**POST api/PassportMO/LpuPeriodOMS**

Добавление периода по ОМС. Сначала вызываем метод получения периодов по ОМС. Добавляемый период не должен пересекаться по времени с полученными периодами.

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. * LpuPeriodOMS\_begDate (D, O) – Дата включения * LpuPeriodOMS\_endDate (D, H) – Дата исключения |
| **Ответ** | **Успешный ответ**   * LpuPeriodOMS\_id (N, O) – Идентификатор периода по ОМС |

* + 1. Изменение периода по ОМС

**PUT api/PassportMO/LpuPeriodOMS**

Изменение периода по ОМС.

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuPeriodOMS\_id (N, O) – Идентификатор периода по ОМС LpuPeriodOMS\_begDate (D, H) – Дата включения * LpuPeriodOMS\_endDate (D, H) – Дата исключения |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0. Измененный период не должен пересекаться по времени с другими периодами по МО. |

* + 1. Получение списка периодов по ОМС в МО

**GET api/PassportMO/LpuPeriodOMSlistByMo**

Получение списка периодов по ОМС в МО

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. |
| **Ответ** | **Успешный ответ** – список периодов по ОМС:   * LpuPeriodOMS\_id (N, O) – Идентификатор периода по ОМС |

* + 1. Получение атрибутов периода по ОМС по идентификатору

**GET api/PassportMO/LpuPeriodOMSbyId**

Получение атрибутов периода по ОМС по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuPeriodOMS\_id (N, O) – Идентификатор периода по ОМС |
| **Ответ** | **Успешный ответ**   * LpuPeriodOMS\_begDate (D, O) – Дата включения * LpuPeriodOMS\_endDate (D, H) – Дата исключения |

* + 1. Удаление периода по ОМС

**DELETE api/PassportMO/LpuPeriodOMS**

Удаление периода по ОМС.

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuPeriodOMS\_id (N, O) – Идентификатор периода по ОМС |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление периода по ЛЛО для МО

**POST api/PassportMO/LpuPeriodDLO**

Добавление периода по ЛЛО. Сначала вызываем метод получения периодов по ЛЛО. Добавляемый период не должен пересекаться по времени с полученными периодами.

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. * LpuPeriodDLO\_begDate (D, O) – Дата включения * LpuPeriodDLO\_endDate (D, H) – Дата исключения |
| **Ответ** | **Успешный ответ**   * LpuPeriodDLO\_id (N, O) – Идентификатор периода по ЛЛО |

* + 1. Изменение периода по ЛЛО для МО

**PUT api/PassportMO/LpuPeriodDLO**

Изменение периода по ЛЛО.

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuPeriodDLO\_id (N, O) – Идентификатор периода по ЛЛО * LpuPeriodDLO\_begDate (D, H) – Дата включения * LpuPeriodDLO\_endDate (D, H) – Дата исключения |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0. Измененный период не должен пересекаться по времени с другими периодами по МО. |

* + 1. Получение списка периодов по ЛЛО в МО

**GET api/PassportMO/LpuPeriodDLObyMo**

Получение списка периодов по ЛЛО в МО

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. |
| **Ответ** | **Успешный ответ** – список периодов по ЛЛО:   * LpuPeriodDLO\_id (N, O) – Идентификатор периода по ЛЛО |

* + 1. Получение атрибутов периода по ЛЛО по идентификатору

**GET api/PassportMO/LpuPeriodDLObyId**

Получение атрибутов периода по ЛЛО по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuPeriodDLO\_id (N, O) – Идентификатор периода по ЛЛО |
| **Ответ** | **Успешный ответ**   * LpuPeriodDLO\_begDate (D, O) – Дата включения * LpuPeriodDLO\_endDate (D, H) – Дата исключения |

* + 1. Удаление периода по ЛЛО

**DELETE api/PassportMO/LpuPeriodDLO**

Удаление периода по ЛЛО.

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuPeriodDLO\_id (N, O) – Идентификатор периода по ЛЛО |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление информационной системы

**POST api/PassportMO/MOInfoSys**

Добавление информационой системы.

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. * DInfSys\_id (N, H) - идентификатор типа ИС. Значение справочника fed.DInfSys * MOInfoSys\_Cost (N, H) – стоимость ИС в рублях. Используется формат денег: 2 знака после запятой и выделение разрядности. * MOInfoSys\_CostYear (N, H) - стоимость сопровождения ИС в год в рублях. Используется формат денег: 2 знака в дробной части и выделение разрядности. * MOInfoSys\_IsMainten (N, О) – признак сопровождения. Значение типа флаг. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * MOInfoSys\_Name (T,О) – Наименование ИС. * MOInfoSys\_NameDeveloper (T, О) – Наименование разработчика * MOInfoSys\_IntroDT (D, О) – Дата внедрения. |
| **Ответ** | **Успешный ответ**   * MOInfoSys\_id – идентификатор ИС |

* + 1. Изменение информационной системы

**PUT api/PassportMO/MOInfoSys**

Изменение информационой системы.

|  |  |
| --- | --- |
| **Входящие параметры** | * MOInfoSys\_id – идентификатор ИС * DInfSys\_id (N, H) - идентификатор типа ИС. Значение справочника fed.DInfSys * MOInfoSys\_Cost (N, H) – стоимость ИС в рублях. Используется формат денег: 2 знака после запятой и выделение разрядности. * MOInfoSys\_CostYear (N, H) - стоимость сопровождения ИС в год в рублях. Используется формат денег: 2 знака в дробной части и выделение разрядности. * MOInfoSys\_IsMainten (N, H) – признак сопровождения. Значение: false или true * MOInfoSys\_Name (T, H) – Наименование ИС. * MOInfoSys\_NameDeveloper (T, H) – Наименование разработчика * MOInfoSys\_IntroDT (D, H) – Дата внедрения. |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0. |

* + 1. Получение списка информационных систем по МО

**GET api/PassportMO/MOInfoSysListByMo**

Получение списка информационных систем

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. |
| **Ответ** | **Успешный ответ -** список:   * MOInfoSys\_id – идентификатор ИС |

* + 1. Получение атрибутов информационной системы по идентификатору

**GET api/PassportMO/MOInfoSysById**

Получение атрибутов информационной системы по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * MOInfoSys\_id – идентификатор ИС |
| **Ответ** | **Успешный ответ**   * DInfSys\_id (N, H) - идентификатор типа ИС. Значение справочника fed.DInfSys * MOInfoSys\_Cost (N, H) – стоимость ИС в рублях. Используется формат денег: 2 знака после запятой и выделение разрядности. * MOInfoSys\_CostYear (N, H) - стоимость сопровождения ИС в год в рублях. Используется формат денег: 2 знака в дробной части и выделение разрядности. * MOInfoSys\_IsMainten (N, О) – признак сопровождения. Значение типа флаг. * MOInfoSys\_Name (T,О) – Наименование ИС. * MOInfoSys\_NameDeveloper (T, О) – Наименование разработчика * MOInfoSys\_IntroDT (D, О) – Дата внедрения. |

* + 1. Удаление информационной системы

**DELETE api/PassportMO/MOInfoSys**

Удаление информационой системы.

|  |  |
| --- | --- |
| **Входящие параметры** | * MOInfoSys\_id – идентификатор ИС |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление специализации организации

**POST api/PassportMO/SpecializationMO**

Добавление специализации организации.

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. * Mkb10Code\_cid (N, O) – Идентификатор класса по МКБ-10. Значение из справочника fed.Mkb10Code * SpecializationMO\_MedProfile (T, O) – Медицинский профиль * SpecializationMO\_IsDepAftercare (N, O) – Наличие отделения долечивания. Значение типа флаг. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * LpuLicence\_id (N, H) – идентификатор лицензии МО |
| **Ответ** | **Успешный ответ**   * SpecializationMO\_id (N, O) – Идентификатор специализации организации |

* + 1. Изменение специализации организации

**PUT api/PassportMO/SpecializationMO**

Изменение специализации организации.

|  |  |
| --- | --- |
| **Входящие параметры** | * SpecializationMO\_id (N, O) – Идентификатор специализации организации * Mkb10Code\_cid (N, H) – Идентификатор класса по МКБ-10. Значение из справочника fed.Mkb10Code * SpecializationMO\_MedProfile (T, H) – Медицинский профиль * SpecializationMO\_IsDepAftercare (N, H) – Наличие отделения долечивания. Значение типа флаг. * LpuLicence\_id (N, H) – идентификатор лицензии МО |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0. |

* + 1. Получение списка специализаций по МО

**GET api/PassportMO/SpecializationMOListByMO**

Получение списка специализаций по МО

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. |
| **Ответ** | **Успешный ответ** – список:   * SpecializationMO\_id (N, O) – Идентификатор специализации организации |

* + 1. Получение атрибутов специализации по идентификатору

**GET api/PassportMO/SpecializationMOById**

Получение атрибутов специализации по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * SpecializationMO\_id (N, O) – Идентификатор специализации организации |
| **Ответ** | **Успешный ответ**   * Mkb10Code\_cid (N, O) – Идентификатор класса по МКБ-10. Значение из справочника fed.Mkb10Code * SpecializationMO\_MedProfile (T, O) – Медицинский профиль * SpecializationMO\_IsDepAftercare (N, O) – Наличие отделения долечивания. Значение типа флаг. * LpuLicence\_id (N, H) – идентификатор лицензии МО |

* + 1. Удаление специализации организации

**DELETE api/PassportMO/SpecializationMO**

Удаление специализации организации.

|  |  |
| --- | --- |
| **Входящие параметры** | * SpecializationMO\_id (N, O) – Идентификатор специализации организации |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление лицензии МО

**POST api/PassportMO/LpuLicence**

Добавление лицензии МО

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. * LpuLicence\_Ser (T, H) – Серия лицензии * LpuLicence\_RegNum (T, H) – регистрационный номер * LpuLicence\_Num (T, O) - Номер лицензии * Org\_id (N,О) – Выдавшая организация. Значение из справочника dbo.Org. * LpuLicence\_setDate (D, O) – Дата выдачи. * LpuLicence\_begDate (D, H) – Начало действия лицензии * LpuLicence\_endDate (D, H) – Окончание действия лицензии |
| **Ответ** | **Успешный ответ**   * LpuLicence\_id (N, O) – Идентификатор лицензии МО |

* + 1. Изменение лицензии МО

**PUT api/PassportMO/LpuLicence**

Изменение лицензии МО

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuLicence\_id (N, O) – Идентификатор лицензии МО * LpuLicence\_Ser (T, H) – Серия лицензии * LpuLicence\_RegNum (T, H) – регистрационный номер * LpuLicence\_Num (T, H) - Номер лицензии * Org\_id (N, H) – Выдавшая организация. Значение из справочника dbo.Org. * LpuLicence\_setDate (D, O) – Дата выдачи. * LpuLicence\_begDate (D, H) – Начало действия лицензии * LpuLicence\_endDate (D, H) – Окончание действия лицензии |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0. |

* + 1. Получение лицензии по МО, номеру лицензии, выдавшей организации и дате выдачи

**GET api/PassportMO/LpuLicenceByMOLicenceNumSetOrgSetDate**

Получение лицензии по МО, номеру лицензии, выдавшей организации и дате выдачи

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. * LpuLicence\_Num (T, O) - Номер лицензии * Org\_id (N, O) – Выдавшая организация. Значение из справочника dbo.Org. * LpuLicence\_setDate (D, H) – Дата выдачи. |
| **Ответ** | **Успешный ответ**:   * LpuLicence\_id (N, O) – Идентификатор лицензии МО |

* + 1. Получение списка лицензий по МО

**GET api/PassportMO/LpuLicenceListByMO**

Получение списка лицензий по МО

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. |
| **Ответ** | **Успешный ответ** – список:   * LpuLicence\_id (N, O) – Идентификатор лицензии МО |
| **Пример** | Запрос  Ответ |

* + 1. Получение атрибутов лицензии МО по идентификатору

**GET api/PassportMO/LpuLicenceById**

Получение атрибутов лицензии МО по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuLicence\_id (N, O) – Идентификатор лицензии МО |
| **Ответ** | **Успешный ответ**   * LpuLicence\_Ser (T, H) – Серия лицензии * LpuLicence\_RegNum (T, H) – регистрационный номер * LpuLicence\_Num (T, O) - Номер лицензии * Org\_id (N,О) – Выдавшая организация. Значение из справочника dbo.Org. * LpuLicence\_setDate (D, O) – Дата выдачи. * LpuLicence\_begDate (D, H) – Начало действия лицензии * LpuLicence\_endDate (D, H) – Окончание действия лицензии |

* + 1. Удаление лицензии МО

**DELETE api/PassportMO/LpuLicence**

Удаление лицензии МО

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuLicence\_id (N, O) – Идентификатор лицензии МО |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление вида лицензии по профилю

**POST api/PassportMO/LpuLicenceProfile**

Добавление вида лицензии по профилю

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuLicence\_id (N, O) – Идентификатор лицензии МО. * LpuLicenceProfileType\_id (N, O) – Идентификатор вида лицензий по профилю. Значения из справочника fed.LpuLicenceProfileType |
| **Ответ** | **Успешный ответ**   * LpuLicenceProfile\_id (N, O) – Идентификатор профиля лицензии |

* + 1. Изменение вида лицензии по профилю

**PUT api/PassportMO/LpuLicenceProfile**

Изменение вида лицензии по профилю

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuLicenceProfile\_id (N, O) – Идентификатор профиля лицензии * LpuLicenceProfileType\_id (N, H) – Идентификатор вида лицензий по профилю. Значения из справочника fed.LpuLicenceProfileType |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0. |

* + 1. Получение списка видов профилей по лицензии

**GET api/PassportMO/LpuLicenceProfileListByLicence**

Получение списка видов профилей по лицензии

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuLicence\_id (N, O) – Идентификатор лицензии МО. |
| **Ответ** | **Успешный ответ** - список:   * LpuLicenceProfile\_id (N, O) – Идентификатор профиля лицензии * LpuLicenceProfileType\_id (N, O) – Идентификатор вида лицензий по профилю. Значения из справочника fed.LpuLicenceProfileType |

* + 1. Удаление вида лицензии по профилю

**DELETE api/PassportMO/LpuLicenceProfile**

Удаление вида лицензии по профилю

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuLicenceProfile\_id (N, O) – Идентификатор профиля лицензии |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление операции, проведенной с лицензией

**POST api/PassportMO/LpuLicenceOperationLink**

Добавление операции, проведенной с лицензией.

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuLicence\_id (N, O) – Идентификатор лицензии МО. * LpuLicenceOperationLink\_Date (D,О) – Дата операции * LicsOperation\_id (N, O) – Идентификатор наименования операции. Значение из справочника fed.LicsOperation |
| **Ответ** | **Успешный ответ**   * LpuLicenceOperationLink\_id (N, O) – Идентификатор операции по лицензии |

* + 1. Изменение операции, проведенной с лицензией

**PUT api/PassportMO/LpuLicenceOperationLink**

Изменение операции, проведенной с лицензией.

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuLicenceOperationLink\_id (N, O) – Идентификатор операции по лицензии * LpuLicenceOperationLink\_Date (D, H) – Дата операции * LicsOperation\_id (N, H) – Идентификатор наименования операции. Значение из справочника fed.LicsOperation |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0. |

* + 1. Получение списка операций, проведенных с лицензией

**GET api/PassportMO/LpuLicenceOperationLinkListByLicence**

Получение списка операций, проведенных с лицензией.

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuLicence\_id (N, O) – Идентификатор лицензии МО. |
| **Ответ** | **Успешный ответ** - список:   * LpuLicenceOperationLink\_id (N, O) – Идентификатор операции по лицензии * LpuLicenceOperationLink\_Date (D,О) – Дата операции * LicsOperation\_id (N, O) – Идентификатор наименования операции. Значение из справочника fed.LicsOperation} |

* + 1. Удаление операции, проведенной с лицензией

**DELETE api/PassportMO/LpuLicenceOperationLink**

Удаление операции, проведенной с лицензией

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuLicenceOperationLink\_id (N, O) – Идентификатор операции по лицензии |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление медтехнологии

**POST api/PassportMO/MedTechnology**

Добавление медтехнологии в паспорте МО

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. * MedTechnology\_Name (T, O) – Наименование медицинской технологии * TechnologyClass\_id (N,О) – Идентификатор класса технологии. Значение из справочника passport.TechnologyClass * LpuBuildingPass\_id (N,О) – Идентификатор здания. (Значение из списка, получаемого по методу GET api/PassportMO/LpuBuildingPassListByMO) |
| **Ответ** | **Успешный ответ**   * MedTechnology\_id (N, O) – Идентификатор медицинской технологии |

* + 1. Изменение медтехнологии

**PUT api/PassportMO/MedTechnology**

Изменение медтехнологии в паспорте МО

|  |  |
| --- | --- |
| **Входящие параметры** | * MedTechnology\_id (N, O) – Идентификатор медицинской технологии * MedTechnology\_Name (T, H) – Наименование медицинской технологии * TechnologyClass\_id (N, H) – Идентификатор класса технологии. Значение из справочника passport.TechnologyClass * LpuBuildingPass\_id (N, H) – Идентификатор здания. (Значение из списка, получаемого по методу GET api/PassportMO/LpuBuildingPassListByMO) |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0. |

* + 1. Получение списка медтехнологий по МО

**GET api/PassportMO/MedTechnologyListByMO**

Получение списка медтехнологий по МО

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. |
| **Ответ** | **Успешный ответ** – список:   * MedTechnology\_id (N, O) – Идентификатор медицинской технологии |

* + 1. Получение списка медтехнологий по идентификатору здания

**GET api/PassportMO/MedTechnologyListByLpuBuildingPass**

Получение списка медтехнологий по идентификатору здания

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuBuildingPass\_id (N, O) – Идентификатор здания. |
| **Ответ** | **Успешный ответ** – список:   * MedTechnology\_id (N, O) – Идентификатор медицинской технологии |

* + 1. Получение атрибутов медтехнологии по идентификатору

**GET api/PassportMO/MedTechnologyById**

Получение атрибутов медтехнологии по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * MedTechnology\_id (N, O) – Идентификатор медицинской технологии |
| **Ответ** | **Успешный ответ**   * MedTechnology\_Name (T, O) – Наименование медицинской технологии * TechnologyClass\_id (N,О) – Идентификатор класса технологии. Значение из справочника passport.TechnologyClass * LpuBuildingPass\_id (N,О) – Идентификатор здания. (Значение из списка, получаемого по методу GET api/PassportMO/LpuBuildingPassListByMO) |

* + 1. Удаление медтехнологии

**DELETE api/PassportMO/MedTechnology**

Удаление медтехнологии в паспорте МО

|  |  |
| --- | --- |
| **Входящие параметры** | * MedTechnology\_id (N, O) – Идентификатор медицинской технологии |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление медицинской услуги в паспорт МО

**POST api/PassportMO/MedUsluga**

Добавление медицинской услуги в паспорт МО

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. * MedUsluga\_LicenseNum (T, O) – Номер лицензии * DUslugi\_id (N, O) – Идентификатор наименования медуслуги. Значение из справочника fed.DUslugi |
| **Ответ** | **Успешный ответ**   * MedUsluga\_id (N, O) – Идентификатор медицинской услуги |

* + 1. Изменение медицинской услуги в паспорте МО

**PUT api/PassportMO/MedUsluga**

Изменение медицинской услуги в паспорте МО

|  |  |
| --- | --- |
| **Входящие параметры** | * MedUsluga\_id (N, O) – Идентификатор медицинской услуги MedUsluga\_LicenseNum (T, H) – Номер лицензии * DUslugi\_id (N, H) – Идентификатор наименования медуслуги. Значение из справочника fed.DUslugi |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0. |

* + 1. Получение списка медицинских услуг по МО

**GET api/PassportMO/MedUslugaListByMO**

Получение списка медицинских услуг по МО

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. |
| **Ответ** | **Успешный ответ** – список:   * MedUsluga\_id (N, O) – Идентификатор медицинской услуги |

* + 1. Получение атрибутов медицинской услуги по идентификатору

**GET api/PassportMO/MedUslugaById**

Получение атрибутов медицинской услуги по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * MedUsluga\_id (N, O) – Идентификатор медицинской услуги |
| **Ответ** | **Успешный ответ**   * MedUsluga\_LicenseNum (T, O) – Номер лицензии * DUslugi\_id (N, O) – Идентификатор наименования медуслуги. Значение из справочника fed.DUslugi |

* + 1. Удаление медицинской услуги из паспорта МО

**DELETE api/PassportMO/MedUsluga**

Удаление медицинской услуги из паспорта МО

|  |  |
| --- | --- |
| **Входящие параметры** | * MedUsluga\_id (N, O) – Идентификатор медицинской услуги |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление направления оказания медицинской помощи

**POST api/PassportMO/UslugaComplexLpu**

Добавление направления оказания медицинской помощи

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. * UslugaComplex\_id (N,О) – Идентификатор услуги. Значение из справочника dbo.UslugaComplex * UslugaComplexLpu\_begDate (D, O) - Дата начала оказания услуги * UslugaComplexLpu\_endDate (D, H) - Дата окончания оказания услуги |
| **Ответ** | **Успешный ответ**   * UslugaComplexLpu\_id (N,О) – Идентификатор направления оказания медицинской помощи. |

* + 1. Изменение направления оказания медицинской помощи

**PUT api/PassportMO/UslugaComplexLpu**

Изменение направления оказания медицинской помощи

|  |  |
| --- | --- |
| **Входящие параметры** | * UslugaComplexLpu\_id (N,О) – Идентификатор направления оказания медицинской помощи. * UslugaComplex\_id (N, H) – Идентификатор услуги. Значение из справочника dbo.UslugaComplex * UslugaComplexLpu\_begDate (D, H) - Дата начала оказания услуги * UslugaComplexLpu\_endDate (D, H) - Дата окончания оказания услуги |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0. |

* + 1. Получение списка направлений оказания медицинской помощи

**GET api/PassportMO/UslugaComplexLpuListByMO**

Получение списка направлений оказания медицинской помощи

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. |
| **Ответ** | **Успешный ответ** – список:   * UslugaComplexLpu\_id (N, О) – Идентификатор направления оказания медицинской помощи. |

* + 1. Получение атрибутов направления оказания медицинской помощи по идентификатору

**GET api/PassportMO/UslugaComplexLpuById**

Получение атрибутов направления оказания медицинской помощи по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * UslugaComplexLpu\_id (N, О) – Идентификатор направления оказания медицинской помощи. |
| **Ответ** | **Успешный ответ**   * UslugaComplex\_id (N,О) – Идентификатор услуги. Значение из справочника dbo.UslugaComplex * UslugaComplexLpu\_begDate (D, O) - Дата начала оказания услуги * UslugaComplexLpu\_endDate (D, H) - Дата окончания оказания услуги |

* + 1. Удаление направления оказания медицинской помощи

**DELETE api/PassportMO/UslugaComplexLpu**

Удаление направления оказания медицинской помощи

|  |  |
| --- | --- |
| **Входящие параметры** | * UslugaComplexLpu\_id (N,О) – Идентификатор направления оказания медицинской помощи. |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление объекта/места использования природного лечебного фактора

**POST api/PassportMO/PlfObjects**

Добавление объекта/места использования природного лечебного фактора

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. * PlfObjects\_id (N, O) – Идентификатор наименования объекта. Значение из справочника fed.PlfObjects. * PlfObjectCount\_Count (N, O) – Количество мест/объектов |
| **Ответ** | **Успешный ответ**   * PlfObjectCount\_id (N, O) – Идентификатор направления оказания медицинской помощи. |

* + 1. Изменение объекта/места использования природного лечебного фактора

**PUT api/PassportMO/PlfObjects**

Изменение объекта/места использования природного лечебного фактора

|  |  |
| --- | --- |
| **Входящие параметры** | * PlfObjectCount\_id (N, O) – Идентификатор направления оказания медицинской помощи. * PlfObjects\_id (N, H) – Идентификатор наименования объекта. Значение из справочника fed.PlfObjects. * PlfObjectCount\_Count (N, H) – Количество мест/объектов |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0. |

* + 1. Получение списка объектов/мест использования природного лечебного фактора по МО

**GET api/PassportMO/PlfObjectsListByMO**

Получение списка объектов/мест использования природного лечебного фактора

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. |
| **Ответ** | **Успешный ответ** - список:   * PlfObjectCount\_id (N, O) – Идентификатор направления оказания медицинской помощи. |

* + 1. Получение атрибутов объекта/места использования природного лечебного фактора по идентификатору

**GET api/PassportMO/PlfObjectsById**

Получение атрибутов объекта/места использования природного лечебного фактора по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * PlfObjectCount\_id (N, O) – Идентификатор направления оказания медицинской помощи. |
| **Ответ** | **Успешный ответ**   * PlfObjects\_id (N, O) – Идентификатор наименования объекта. Значение из справочника fed.PlfObjects. * PlfObjectCount\_Count (N, O) – Количество мест/объектов |

* + 1. Удаление объекта/места использования природного лечебного фактора

**DELETE api/PassportMO/PlfObjects**

Удаление объекта/места использования природного лечебного фактора

|  |  |
| --- | --- |
| **Входящие параметры** | * PlfObjectCount\_id (N, O) – Идентификатор направления оказания медицинской помощи. |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление периода функционирования МО

**POST api/PassportMO/FunctionTime**

Добавление периода функционирования МО

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. * InstitutionFunction\_id (N, O) – Идентификатор периода функционирования учреждения. Значение из справочника passport.InstitutionFunction * FunctionTime\_begDate (D, O) – Дата начала периода функционирования учреждения * FunctionTime\_endDate (D, H) – Дата окончания периода функционирования учреждения |
| **Ответ** | **Успешный ответ**   * FunctionTime\_id (N, O) – Идентификатор периода функционирования учреждения |

* + 1. Изменение периода функционирования МО

**PUT api/PassportMO/FunctionTime**

Изменение периода функционирования МО

|  |  |
| --- | --- |
| **Входящие параметры** | * FunctionTime\_id (N, O) – Идентификатор периода функционирования учреждения * InstitutionFunction\_id (N, H) – Идентификатор периода функционирования учреждения. Значение из справочника passport.InstitutionFunction * FunctionTime\_begDate (D, H) – Дата начала периода функционирования учреждения * FunctionTime\_endDate (D, H) – Дата окончания периода функционирования учреждения |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0. |

* + 1. Получение списка периодов функционирования по МО

**GET api/PassportMO/FunctionTimeListByMO**

Добавление списка периодов функционирования по МО

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. |
| **Ответ** | **Успешный ответ** – список:   * FunctionTime\_id (N, O) – Идентификатор периода функционирования учреждения |

* + 1. Получение атрибутов периода функционирования по идентификатору

**GET api/PassportMO/FunctionTimeById**

Получение атрибутов периода функционирования по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * FunctionTime\_id (N, O) – Идентификатор периода функционирования учреждения |
| **Ответ** | **Успешный ответ**   * InstitutionFunction\_id (N, O) – Идентификатор периода функционирования учреждения. Значение из справочника passport.InstitutionFunction * FunctionTime\_begDate (D, O) – Дата начала периода функционирования учреждения * FunctionTime\_endDate (D, H) – Дата окончания периода функционирования учреждения |

* + 1. Удаление периода функционирования МО

**DELETE api/PassportMO/FunctionTime s**

Удаление периода функционирования МО

|  |  |
| --- | --- |
| **Входящие параметры** | * FunctionTime\_id (N, O) – Идентификатор периода функционирования учреждения |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление питания

**POST api/PassportMO/Pitan**

Добавление питания.

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. * PitanForm\_id (N, O) – Идентификатор формы питания. Значение из справочника  [fed.PitanForm](http://fixuploader.swn.local/dbspr/index/584414643) * VidPitan\_id (N, O) – Идентификатор вида питания. Значение из справочника  [fed.VidPitan](http://fixuploader.swn.local/dbspr/index/616414757) * PitanCnt\_id (N, O) – Идентификатор кратности питания. Значение из справочника  [fed.PitanCnt](http://fixuploader.swn.local/dbspr/index/552414529) |
| **Ответ** | **Успешный ответ**   * PitanFormTypeLink\_id (N, O) – Идентификатор таблицы связей питания   Если уже существует аналогичное питание, то возвращает ошибку. |

* + 1. Изменение питания

**PUT api/PassportMO/Pitan**

Изменение питания

|  |  |
| --- | --- |
| **Входящие параметры** | * PitanFormTypeLink\_id (N, O) – Идентификатор таблицы связей питания * PitanForm\_id (N, H) – Идентификатор формы питания. Значение из справочника  [fed.PitanForm](http://fixuploader.swn.local/dbspr/index/584414643) * VidPitan\_id (N, H) – Идентификатор вида питания. Значение из справочника  [fed.VidPitan](http://fixuploader.swn.local/dbspr/index/616414757) * PitanCnt\_id (N, H) – Идентификатор кратности питания. Значение из справочника  [fed.PitanCnt](http://fixuploader.swn.local/dbspr/index/552414529) |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0.  Если уже существует аналогичное питание, то возвращает ошибку. |

* + 1. Получение списка питаний по МО

**GET api/PassportMO/PitanListByMO**

Добавление списка питаний по МО.

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. |
| **Ответ** | **Успешный ответ** - список:   * PitanFormTypeLink\_id (N, O) – Идентификатор таблицы связей питания |

* + 1. Получение атрибутов питания по идентификатору

**GET api/PassportMO/PitanById**

Получение атрибутов питания по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * PitanFormTypeLink\_id (N, O) – Идентификатор таблицы связей питания |
| **Ответ** | **Успешный ответ**   * PitanForm\_id (N, O) – Идентификатор формы питания. Значение из справочника fed.PitanForm * VidPitan\_id (N, O) – Идентификатор вида питания. Значение из справочника fed.VidPitan * PitanCnt\_id (N, O) – Идентификатор кратности питания. Значение из справочника fed.PitanCnt |

* + 1. Удаление питания

**DELETE api/PassportMO/Pitan**

Удаление питания

|  |  |
| --- | --- |
| **Входящие параметры** | * PitanFormTypeLink\_id (N, O) – Идентификатор таблицы связей питания |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление природного лечебного фактора

**POST api/PassportMO/Plf**

Добавление природного лечебного фактора

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. * DocTypeUsePlf\_id (N, O) – Документ. Значение из справочника  [fed.DocTypeUsePlf](http://fixuploader.swn.local/dbspr/index/1355893389) * PlfType\_id (N, O) – Идентификатор типа природного лечебного фактора. Значение из справочника  [fed.PlfType](http://fixuploader.swn.local/dbspr/index/680414985) * Plf\_id (N, O) – Идентификатор наименования природного лечебного фактора. Значение из справочника  [fed.Plf](http://fixuploader.swn.local/dbspr/index/648414871) * PlfDocTypeLink\_Num (T, O) – Номер * PlfDocTypeLink\_BegDT (D, H) – Дата начала действия фактора * PlfDocTypeLink\_EndDT (D, H) – Дата окончания действия фактора * PlfDocTypeLink\_GetDT (D, O) - Дата выдачи документа |
| **Ответ** | **Успешный ответ**   * PlfDocTypeLink\_id (N, O) – Идентификатор таблицы связей природных леебных факторов и МО |

* + 1. Изменение природного лечебного фактора

**PUT api/PassportMO/Plf**

Изменение природного лечебного фактора

|  |  |
| --- | --- |
| **Входящие параметры** | * PlfDocTypeLink\_id (N, O) – Идентификатор таблицы связей природных леебных факторов и МО * DocTypeUsePlf\_id (N, H) – Документ. Значение из справочника  [fed.DocTypeUsePlf](http://fixuploader.swn.local/dbspr/index/1355893389) * PlfType\_id (N, H) – Идентификатор типа природного лечебного фактора. Значение из справочника  [fed.PlfType](http://fixuploader.swn.local/dbspr/index/680414985) * Plf\_id (N, H) – Идентификатор наименования природного лечебного фактора. Значение из справочника  [fed.Plf](http://fixuploader.swn.local/dbspr/index/648414871) * PlfDocTypeLink\_Num (T, H) – Номер * PlfDocTypeLink\_BegDT (D, H) – Дата начала действия фактора * PlfDocTypeLink\_EndDT (D, H) – Дата окончания действия фактора * PlfDocTypeLink\_GetDT (D, H) - Дата выдачи документа |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0. |

* + 1. Получение списка природных лечебных факторов

**GET api/PassportMO/PlfListByMO**

Получение списка природных лечебных факторов

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. |
| **Ответ** | **Успешный ответ** - список:   * PlfDocTypeLink\_id (N, O) – Идентификатор таблицы связей природных леебных факторов и МО |

* + 1. Получение атрибутов природного лечебного фактора по идентификатору

**GET api/PassportMO/PlfById**

Получение атрибутов природного лечебного фактора по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * PlfDocTypeLink\_id (N, O) – Идентификатор таблицы связей природных леебных факторов и МО |
| **Ответ** | **Успешный ответ** - список:   * DocTypeUsePlf\_id (N, O) – Документ. Значение из справочника fed.DocTypeUsePlf * PlfType\_id (N, O) – Идентификатор типа природного лечебного фактора. Значение из справочника fed.PlfType * Plf\_id (N, O) – Идентификатор наименования природного лечебного фактора. Значение из справочника fed.Plf * PlfDocTypeLink\_Num (T, O) – Номер * PlfDocTypeLink\_BegDT (D, H) – Дата начала действия фактора * PlfDocTypeLink\_EndDT (D, H) – Дата окончания действия фактора * PlfDocTypeLink\_GetDT (D, O) - Дата выдачи документа |

* + 1. Удаление природного лечебного фактора

**DELETE api/PassportMO/Plf**

Удаление природного лечебного фактора

|  |  |
| --- | --- |
| **Входящие параметры** | * PlfDocTypeLink\_id (N, O) – Идентификатор таблицы связей природных леебных факторов и МО |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление расчетного счета

**POST api/PassportMO/OrgRSchet**

Добавление расчетного счета

|  |  |
| --- | --- |
| **Входящие параметры** | * Org\_id (N,О) - Идентификатор организации. * OrgRSchetType\_id (N, O) – Идентификатор типа расчетного счета. Значение из справочника dbo.OrgRSchetType * OrgRSchet\_RSchet (T, O) – Номер расчетного счета. * OrgBank\_id (N, O) – Идентификатор банка. Значение из справочника dbo.OrgBank * Okv\_id (N, O) – Идентификатор валюты. Значение из справочника dbo.Okv * OrgRSchet\_Name (T, O) – Наименование расчетного счета * OrgRSchet\_begDate (D, O) – Дата открытия расчетного счета * OrgRSchet\_endDate (D, H) – Дата закрытия расчетного счета |
| **Ответ** | **Успешный ответ**   * OrgRSchet\_id (N, O) – Идентификатор расчетного счета. |

* + 1. Изменение расчетного счета

**PUT api/PassportMO/OrgRSchet**

Изменение расчетного счета

|  |  |
| --- | --- |
| **Входящие параметры** | * OrgRSchet\_id (N, O) – Идентификатор расчетного счета. * OrgRSchetType\_id (N, H) – Идентификатор типа расчетного счета. Значение из справочника dbo.OrgRSchetType * OrgRSchet\_RSchet (T, H) – Номер расчетного счета. * OrgBank\_id (N, H) – Идентификатор банка. Значение из справочника dbo.OrgBank * Okv\_id (N, H) – Идентификатор валюты. Значение из справочника dbo.Okv * OrgRSchet\_Name (T, H) – Наименование расчетного счета * OrgRSchet\_begDate (D, H) – Дата открытия расчетного счета * OrgRSchet\_endDate (D, H) – Дата закрытия расчетного счета |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0. |

* + 1. Получение списка расчетных счетов по МО

**GET api/PassportMO/OrgRSchetListByMO**

Получение списка расчетных счетов по МО

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. |
| **Ответ** | **Успешный ответ** – список:   * OrgRSchet\_id (N, O) – Идентификатор расчетного счета. * OrgRSchet\_RSchet (T, O) – Номер расчетного счета. * OrgBank\_id (N, O) – Идентификатор банка. Значение из справочника dbo.OrgBank * Okv\_id (N, O) – Идентификатор валюты. Значение из справочника dbo.Okv * OrgRSchet\_Name (T, O) – Наименование расчетного счета * OrgRSchet\_begDate (D, O) – Дата открытия расчетного счета * OrgRSchet\_endDate (D, H) – Дата закрытия расчетного счета |

* + 1. Получение атрибутов расчетного счета по идентификатору

**GET api/PassportMO/OrgRSchetByMO**

Получение атрибутов расчетного счета по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * OrgRSchet\_id (N, O) – Идентификатор расчетного счета. |
| **Ответ** | **Успешный ответ**   * OrgRSchetType\_id (N, O) – Идентификатор типа расчетного счета. Значение из справочника dbo.OrgRSchetType * OrgRSchet\_RSchet (T, O) – Номер расчетного счета. * OrgBank\_id (N, O) – Идентификатор банка. Значение из справочника dbo.OrgBank * Okv\_id (N, O) – Идентификатор валюты. Значение из справочника dbo.Okv * OrgRSchet\_Name (T, O) – Наименование расчетного счета * OrgRSchet\_begDate (D, O) – Дата открытия расчетного счета * OrgRSchet\_endDate (D, H) – Дата закрытия расчетного счета |

* + 1. Удаление расчетного счета

**DELETE api/PassportMO/OrgRSchet**

Удаление расчетного счета

|  |  |
| --- | --- |
| **Входящие параметры** | * OrgRSchet\_id (N, O) – Идентификатор расчетного счета. |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление КБК для расчетного счета

**POST api/PassportMO/OrgRSchetKBK**

Добавление КБК для расчетного счета

|  |  |
| --- | --- |
| **Входящие параметры** | * OrgRSchet\_id (N, O) – Идентификатор расчетного счета. * OrgRSchet\_KBK (T, O) – КБК организации |
| **Ответ** | **Успешный ответ**   * OrgRSchetKBK\_id (N, O) – Идентификатор привязки КБК к расчетному счету. |

* + 1. Изменение КБК для расчетного счета

**PUT api/PassportMO/OrgRSchetKBK**

Изменение КБК для расчетного счета

|  |  |
| --- | --- |
| **Входящие параметры** | * OrgRSchetKBK\_id (N, O) – Идентификатор привязки КБК к расчетному счету. * OrgRSchet\_KBK (T, H) – КБК организации |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0. |

* + 1. Получение списка КБК по расчетному счету

**GET api/PassportMO/OrgRSchetKBKListByRSchet**

Получение КБК по расчетному счету

|  |  |
| --- | --- |
| **Входящие параметры** | * OrgRSchet\_id (N, O) – Идентификатор расчетного счета. |
| **Ответ** | **Успешный ответ** – список:   * OrgRSchetKBK\_id (N, O) – Идентификатор привязки КБК к расчетному счету. * OrgRSchet\_KBK (T[20];O) – КБК организации. |

* + 1. Удаление КБК для расчетного счета

**DELETE api/PassportMO/OrgRSchetKBK**

Удаление КБК для расчетного счета

|  |  |
| --- | --- |
| **Входящие параметры** | * OrgRSchetKBK\_id (N, O) – Идентификатор привязки КБК к расчетному счету. |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление территории обслуживания

**POST api/PassportMO/OrgServiceTerr**

Добавление территории обслуживания

|  |  |
| --- | --- |
| **Входящие параметры** | * Org\_id (N,О) - Идентификатор организации. * KLCountry\_id (N, О) – Идентификатор страны. Значение из справочника dbo.KLCountry * KLRgn\_id (N, О) – Идентификатор региона. Значение из справочника dbo.KLArea * KLSubRgn\_id (N, H) – Идентификатор района. Значение из справочника dbo.KLArea * KLCity\_id (N, H) – Идентификатор города. Значение из справочника dbo.KLArea * KLTown\_id (N, H) – Идентификатор нас. пункта. Значение из справочника dbo.KLArea * KLAreaType\_id (N, H) – Идентификатор типа населенного пункта. Значение из справочника dbo.KLAreaType |
| **Ответ** | **Успешный ответ**   * OrgServiceTerr\_id (N, O) – Идентификатор территории обслуживания |

* + 1. Изменение территории обслуживания

**PUT api/PassportMO/OrgServiceTerr**

Изменение территории обслуживания

|  |  |
| --- | --- |
| **Входящие параметры** | * OrgServiceTerr\_id (N, O) – Идентификатор территории обслуживания * KLCountry\_id (N, H) – Идентификатор страны. Значение из справочника dbo.KLCountry * KLRgn\_id (N, H) – Идентификатор региона. Значение из справочника dbo.KLArea * KLSubRgn\_id (N, H) – Идентификатор района. Значение из справочника dbo.KLArea * KLCity\_id (N, H) – Идентификатор города. Значение из справочника dbo.KLArea * KLTown\_id (N, H) – Идентификатор нас. пункта. Значение из справочника dbo.KLArea * KLAreaType\_id (N, H) – Идентификатор типа населенного пункта. Значение из справочника dbo.KLAreaType |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0. |

* + 1. Получение списка территорий обслуживания

**GET api/PassportMO/OrgServiceTerrListByMO**

Получение списка территорий обслуживания

|  |  |
| --- | --- |
| **Входящие параметры** | * Org\_id (N,О) - Идентификатор организации. |
| **Ответ** | **Успешный ответ** – список:   * OrgServiceTerr\_id (N, O) – Идентификатор территории обслуживания. |

* + 1. Получение атрибутов территории обслуживания по идентификатору

**GET api/PassportMO/OrgServiceTerrById**

Получение атрибутов территории обслуживания по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * OrgServiceTerr\_id (N, O) – Идентификатор территории обслуживания. |
| **Ответ** | **Успешный ответ**   * KLCountry\_id (N, О) – Идентификатор страны. Значение из справочника dbo.KLCountry * KLRgn\_id (N, О) – Идентификатор региона. Значение из справочника dbo.KLArea * KLSubRgn\_id (N, H) – Идентификатор района. Значение из справочника dbo.KLArea * KLCity\_id (N, H) – Идентификатор города. Значение из справочника dbo.KLArea * KLTown\_id (N, H) – Идентификатор нас. пункта. Значение из справочника dbo.KLArea * KLAreaType\_id (N, H) – Идентификатор типа населенного пункта. Значение из справочника dbo.KLAreaType |

* + 1. Удаление территории обслуживания

**DELETE api/PassportMO/OrgServiceTerr**

Удаление территории обслуживания

|  |  |
| --- | --- |
| **Входящие параметры** | * OrgServiceTerr\_id (N, O) – Идентификатор территории обслуживания |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление руководства организации

**POST api/PassportMO/OrgHead**

Добавление руководства организации

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. * Person\_id (N, O) – Идентификатор человека. Сначала вызывается метод поиск человека. Если не найден, то вызывается метод добавления человека. * OrgHeadPost\_id (N, О) – Идентификатор руководящей должности. Значение из справочника dbo.OrgHeadPost * OrgHead\_Phone (T, H) – Телефон. * OrgHead\_Fax (T, H) – Факс. * OrgHead\_Email (T, H) – Электронная почта. * OrgHead\_CommissDate (D, H) – Дата приказа о назначении |
| **Ответ** | **Успешный ответ**   * OrgHead\_id (N, O) – Идентификатор руководящей единицы организации. |

* + 1. Изменение руководства организации

**PUT api/PassportMO/OrgHead**

Изменение руководства организации

|  |  |
| --- | --- |
| **Входящие параметры** | * OrgHead\_id (N, O) – Идентификатор руководящей единицы организации. * OrgHeadPost\_id (N, H) – Идентификатор руководящей должности. Значение из справочника dbo.OrgHeadPost * OrgHead\_Phone (T, H) – Телефон. * OrgHead\_Fax (T, H) – Факс. * OrgHead\_Email (T, H) – Электронная почта. * OrgHead\_CommissDate (D, H) – Дата приказа о назначении |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0. |

* + 1. Получение списка руководства по МО

**GET api/PassportMO/OrgHeadListByMO**

Получение списка руководства по МО

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. |
| **Ответ** | **Успешный ответ** – список:   * OrgHead\_id (N, O) – Идентификатор руководящей единицы организации. |

* + 1. Получение руководства организации по идентификатору

**GET api/PassportMO/OrgHeadById**

Получение руководства организации по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * OrgHead\_id (N, O) – Идентификатор руководящей единицы организации. |
| **Ответ** | **Успешный ответ** – список:   * OrgHead\_id (N, O) – Идентификатор руководящей единицы организации. |

* + 1. Удаление руководства организации

**DELETE api/PassportMO/OrgHead**

Удаление руководства организации

|  |  |
| --- | --- |
| **Входящие параметры** | * OrgHead\_id (N, O) – Идентификатор руководящей единицы организации. |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление заезда

**POST api/PassportMO/MOArrival**

Добавление заезда

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. * MOArrival\_EndDT (D, O) – Дата окончания заезда * MOArrival\_CountPerson (N, H) – Количество человек * MOArrival\_TreatDis (N, O) – Длительность заезда |
| **Ответ** | **Успешный ответ**   * MOArrival\_id (N, O) – Идентификатор заезда |

* + 1. Изменение заезда

**PUT api/PassportMO/MOArrival**

Изменение заезда

|  |  |
| --- | --- |
| **Входящие параметры** | * MOArrival\_id (N, O) – Идентификатор заезда * MOArrival\_EndDT (D, H) – Дата окончания заезда * MOArrival\_CountPerson (N, H) – Количество человек * MOArrival\_TreatDis (N, H) – Длительность заезда |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0. |

* + 1. Получение списка заездов по МО

**GET api/PassportMO/MOArrivalListByMO**

Получение списка заездов по МО

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. |
| **Ответ** | **Успешный ответ** – список:   * MOArrival\_id (N, O) – Идентификатор заезда |
| **Пример** | Запрос  Ответ |

* + 1. Получение атрибутов заезда по идентификатору

**GET api/PassportMO/MOArrivalById**

Получение списка заездов по МО

|  |  |
| --- | --- |
| **Входящие параметры** | * MOArrival\_id (N, O) – Идентификатор заезда |
| **Ответ** | **Успешный ответ**   * MOArrival\_EndDT (D, O) – Дата окончания заезда * MOArrival\_CountPerson (N, H) – Количество человек * MOArrival\_TreatDis (N, O) – Длительность заезда |

* + 1. Удаление заезда

**DELETE api/PassportMO/MOArrival**

Удаление заезда

|  |  |
| --- | --- |
| **Входящие параметры** | * MOArrival\_id (N, O) – Идентификатор заезда |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление округа горно-санитарной охраны

**POST api/PassportMO/DisSanProtection**

Добавление округа горно-санитарной охраны

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. * DisSanProtection\_Date (D, O) – Дата документа. * DisSanProtection\_Doc (T[256], O) – Документ. * DisSanProtection\_Num (T[20], O) – Номер документа. * DisSanProtection\_IsProtection (N, O) – Признак наличия округа. Значение типа флаг. Возможные значения: 0 и 1, - где 0 – нет, 1 – да |
| **Ответ** | **Успешный ответ**   * DisSanProtection\_id (N, O) – Идентификатор округа горно-санитарной охраны. |

* + 1. Изменение округа горно-санитарной охраны

**PUT api/PassportMO/DisSanProtection**

Изменение округа горно-санитарной охраны

|  |  |
| --- | --- |
| **Входящие параметры** | * DisSanProtection\_id (N, O) – Идентификатор округа горно-санитарной охраны. * DisSanProtection\_Date (D, H) – Дата документа. * DisSanProtection\_Doc (T[256], H) – Документ. * DisSanProtection\_Num (T[20], H) – Номер документа. * DisSanProtection\_IsProtection (N, H) – Признак наличия округа. Значение типа флаг. |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0. |

* + 1. Получение списка округов горно-санитарной охраны

**GET api/PassportMO/DisSanProtectionListByMO**

Получение списка округов горно-санитарной охраны

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. |
| **Ответ** | **Успешный ответ** – список:   * DisSanProtection\_id (N, O) – Идентификатор округа горно-санитарной охраны. |

* + 1. Получение атрибутов округа горно-санитарной охраны по идентификатору

**GET api/PassportMO/DisSanProtectionById**

Получение атрибутов округа горно-санитарной охраны по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * DisSanProtection\_id (N, O) – Идентификатор округа горно-санитарной охраны. |
| **Ответ** | **Успешный ответ**   * DisSanProtection\_Date (D, O) – Дата документа. * DisSanProtection\_Doc (T[256], O) – Документ. * DisSanProtection\_Num (T[20], O) – Номер документа. * DisSanProtection\_IsProtection (N, O) – Признак наличия округа. Значение типа флаг. |

* + 1. Удаление округа горно-санитарной охраны

**DELETE api/PassportMO/DisSanProtection**

Удаление округа горно-санитарной охраны

|  |  |
| --- | --- |
| **Входящие параметры** | * DisSanProtection\_id (N, O) – Идентификатор округа горно-санитарной охраны. |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление статуса курорта

**POST api/PassportMO/KurortStatusDoc**

Добавление округа горно-санитарной охраны

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. * KurortStatus\_id (N, O) – Идентификатор наименования статуса курорта. Значение из справочника fed.KurortStatus * KurortStatusDoc\_Date (D, O) – Дата документа. * KurortStatusDoc\_Doc (T[256], O) – Документ. * KurortStatusDoc\_Num (T[20], O) – Номер документа. * KurortStatusDoc\_IsStatus (N, O) – Наличие статуса курорта. Значение типа флаг. Возможные значения: 0 и 1, - где 0 – нет, 1 – да |
| **Ответ** | **Успешный ответ**   * KurortStatusDoc\_id (N, O) – Идентификатор статуса курорта. |

* + 1. Изменение статуса курорта

**PUT api/PassportMO/KurortStatusDoc**

Изменение статуса курорта

|  |  |
| --- | --- |
| **Входящие параметры** | * KurortStatusDoc\_id (N, O) – Идентификатор статуса курорта. * KurortStatus\_id (N, H) – Идентификатор наименования статуса курорта. Значение из справочника fed.KurortStatus * KurortStatusDoc\_Date (D, H) – Дата документа. * KurortStatusDoc\_Doc (T[256], H) – Документ. * KurortStatusDoc\_Num (T[20], H) – Номер документа. * KurortStatusDoc\_IsStatus (N, H) – Наличие статуса курорта. Значение типа флаг. |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0. |

* + 1. Получение списка статусов курорта по МО

**GET api/PassportMO/KurortStatusDocListByMO**

Получение списка статусов курорта по МО

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. |
| **Ответ** | **Успешный ответ** - список:   * KurortStatusDoc\_id (N, O) – Идентификатор статуса курорта. |

* + 1. Получение атрубутов статуса курорта по идентификатору

**POST api/PassportMO/KurortStatusDocById**

Получение атрубутов статуса курорта по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * KurortStatusDoc\_id (N, O) – Идентификатор статуса курорта. |
| **Ответ** | **Успешный ответ** - список:   * KurortStatus\_id (N, O) – Идентификатор наименования статуса курорта. Значение из справочника fed.KurortStatus * KurortStatusDoc\_Date (D, O) – Дата документа. * KurortStatusDoc\_Doc (T[256], O) – Документ. * KurortStatusDoc\_Num (T[20], O) – Номер документа. * KurortStatusDoc\_IsStatus (N, O) – Наличие статуса курорта. Значение типа флаг. |

* + 1. Удаление статуса курорта

**DELETE api/PassportMO/KurortStatusDoc**

Удаление статуса курорта

|  |  |
| --- | --- |
| **Входящие параметры** | * KurortStatusDoc\_id (N, O) – Идентификатор статуса курорта. |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление типа курорта

**POST api/PassportMO/KurortTypeLink**

Добавление типа курорта

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. * KurortType\_id (N, O) – Идентификатор наименования типа курорта. Значение из справочника fed.KurortType * KurortTypeLink\_Date (D, O) – Дата документа. * KurortTypeLink\_Doc (T[256], O) – Документ. * KurortTypeLink\_Num (T[20], O) – Номер документа. * KurortTypeLink\_IsKurortTypeLink (N, O) – Наличие типа курорта. Значение типа флаг. Возможные значения: 0 и 1, - где 0 – нет, 1 – да |
| **Ответ** | **Успешный ответ**   * KurortTypeLink\_id (N, O) – Идентификатор типа курорта. |

* + 1. Изменение типа курорта

**PUT api/PassportMO/KurortTypeLink**

Изменение типа курорта

|  |  |
| --- | --- |
| **Входящие параметры** | * KurortTypeLink\_id (N, O) – Идентификатор типа курорта. * KurortType\_id (N, H) – Идентификатор наименования типа курорта. Значение из справочника fed.KurortType * KurortTypeLink\_Date (D, H) – Дата документа. * KurortTypeLink\_Doc (T[256], H) – Документ. * KurortTypeLink\_Num (T[20], H) – Номер документа. * KurortTypeLink\_IsKurortTypeLink (N, H) – Наличие типа курорта. Значение типа флаг. |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0. |

* + 1. Получение списка типов курорта по МО

**GET api/PassportMO/KurortTypeLinkListByMO**

Получение списка типов курорта по МО

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. |
| **Ответ** | **Успешный ответ** – список:   * KurortTypeLink\_id (N, O) – Идентификатор типа курорта. |

* + 1. Получение типа курорта по идентификатору

**GET api/PassportMO/KurortTypeLinkById**

Получение типа курорта по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * KurortTypeLink\_id (N, O) – Идентификатор типа курорта. |
| **Ответ** | **Успешный ответ** – список:   * KurortType\_id (N, O) – Идентификатор наименования типа курорта. Значение из справочника fed.KurortType * KurortTypeLink\_Date (D, O) – Дата документа. * KurortTypeLink\_Doc (T[256], O) – Документ. * KurortTypeLink\_Num (T[20], O) – Номер документа. * KurortTypeLink\_IsKurortTypeLink (N, O) – Наличие типа курорта. Значение типа флаг. |

* + 1. Удаление типа курорта

**DELETE api/PassportMO/KurortTypeLink**

Удаление типа курорта

|  |  |
| --- | --- |
| **Входящие параметры** | * KurortTypeLink\_id (N, O) – Идентификатор типа курорта. |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление объекта инфраструктуры

**POST api/PassportMO/ MOAreaObject**

Добавление объекта инфраструктуры

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. * DObjInfrastructure\_id (N, O) – Идентификатор наименования объекта инфраструктуры. Значение из справочника fed.DObjInfrastructure * MOAreaObject\_Count (N, H) – Количество объектов инфраструктуры. * MOAreaObject\_Member (T[256], O) – Идентификатор участка. |
| **Ответ** | **Успешный ответ**   * MOAreaObject\_id (N, O) – Идентификатор объекта инфраструктуры. |

* + 1. Изменение объекта инфраструктуры

**PUT api/PassportMO/ MOAreaObject**

Изменение объекта инфраструктуры

|  |  |
| --- | --- |
| **Входящие параметры** | * MOAreaObject\_id (N, O) – Идентификатор объекта инфраструктуры. * DObjInfrastructure\_id (N, H) – Идентификатор наименования объекта инфраструктуры. Значение из справочника fed.DObjInfrastructure * MOAreaObject\_Count (N, H) – Количество объектов инфраструктуры. * MOAreaObject\_Member (T[256], H) – Идентификатор участка. |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0. |

* + 1. Получение списка объектов инфраструктуры по МО

**GET api/PassportMO/ MOAreaObject**

Получение списка объектов инфраструктуры по МО

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. |
| **Ответ** | **Успешный ответ -** список:   * MOAreaObject\_id (N, O) – Идентификатор объекта инфраструктуры. |

* + 1. Получение объекта инфраструктуры по идентификатору

**GET api/PassportMO/ MOAreaObject**

Получение объекта инфраструктуры по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * MOAreaObject\_id (N, O) – Идентификатор объекта инфраструктуры. |
| **Ответ** | **Успешный ответ**   * DObjInfrastructure\_id (N, O) – Идентификатор наименования объекта инфраструктуры. Значение из справочника fed.DObjInfrastructure * MOAreaObject\_Count (N, H) – Количество объектов инфраструктуры. * MOAreaObject\_Member (T[256], O) – Идентификатор участка. |

* + 1. Удаление объекта инфраструктуры

**DELETE api/PassportMO/ MOAreaObject**

Удаление объекта инфраструктуры

|  |  |
| --- | --- |
| **Входящие параметры** | * MOAreaObject\_id (N, O) – Идентификатор объекта инфраструктуры. |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление площадки, занимаемой организацией

**POST api/PassportMO/MOArea**

Добавление площадки, занимаемой организацией

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. * MOArea\_Name (T[256], O) – Наименование площадки. * MOArea\_Member (T[256], O) – Идентификатор участка. * MoArea\_Right (T[256], H) – Право на земельный участок. * MoArea\_Space (N,H) - Площадь участка, га (5 знаков в дробной части). * MoArea\_KodTer (T[50], O) – Код территории. * MoArea\_OrgDT (D, H) – Дата организации. * MoArea\_AreaSite (N,H) - Площадь площадки, га (5 знаков в дробной части). * OKATO\_id (N, H) – Идентификатор кода ОКАТО. Значение из справочника nsi.OKATO * Address\_id (N, H) – Идентификатор адреса. Значение ответа при вызове метода создание адреса при ручном вводе. |
| **Ответ** | **Успешный ответ**   * MOArea\_id (N, O) – Идентификатор площадки, занимаемой организацией. |

* + 1. Изменение площадки, занимаемой организацией

**PUT api/PassportMO/MOArea**

Изменение площадки, занимаемой организацией

|  |  |
| --- | --- |
| **Входящие параметры** | * MOArea\_id (N, O) – Идентификатор площадки, занимаемой организацией. * MOArea\_Name (T[256], H) – Наименование площадки. * MOArea\_Member (T[256], H) – Идентификатор участка. * MoArea\_Right (T[256], H) – Право на земельный участок. * MoArea\_Space (N,H) - Площадь участка, га (5 знаков в дробной части). * MoArea\_KodTer (T[50], H) – Код территории. * MoArea\_OrgDT (D, H) – Дата организации. * MoArea\_AreaSite (N,H) - Площадь площадки, га (5 знаков в дробной части). * OKATO\_id (N, H) – Идентификатор кода ОКАТО. Значение из справочника nsi.OKATO * Address\_id (N, H) – Идентификатор адреса. Значение ответа при вызове метода создания/редактирования адреса при ручном вводе. |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0. |

* + 1. Получение списка площадок, занимаемых организацией

**GET api/PassportMO/MOAreaListByMO**

Получение списка площадок, занимаемых организацией

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. |
| **Ответ** | **Успешный ответ** – список:   * MOArea\_id (N, O) – Идентификатор площадки, занимаемой организацией. |

* + 1. Получение идентификатора площадки по наименованию площадки и идентификатору участка

**GET api/PassportMO/MOAreaByNameAndMember**

Получение идентификатора площадки по наименованию площадки и идентификатору участка

|  |  |
| --- | --- |
| **Входящие параметры** | * MOArea\_Name (T[256], O) – Наименование площадки. * MOArea\_Member (T[256], O) – Идентификатор участка. |
| **Ответ** | **Успешный ответ**:   * MOArea\_id (N, O) – Идентификатор площадки, занимаемой организацией. |

* + 1. Получение атрибутов площадки, занимаемой организацией, по идентификатору

**GET api/PassportMO/MOAreaById**

Получение атрибутов площадки, занимаемой организацией, по идентификатору

|  |  |
| --- | --- |
| **Входящие параметры** | * MOArea\_id (N, O) – Идентификатор площадки, занимаемой организацией. |
| **Ответ** | **Успешный ответ**   * MOArea\_Name (T[256], O) – Наименование площадки. * MOArea\_Member (T[256], O) – Идентификатор участка. * MoArea\_Right (T[256], H) – Право на земельный участок. * MoArea\_Space (N,H) - Площадь участка, га (5 знаков в дробной части). * MoArea\_KodTer (T[50], O) – Код территории. * MoArea\_OrgDT (D, H) – Дата организации. * MoArea\_AreaSite (N,H) - Площадь площадки, га (5 знаков в дробной части). * OKATO\_id (N, H) – Идентификатор кода ОКАТО. Значение из справочника nsi.OKATO * Address\_id (N, H) – Идентификатор адреса. Значение ответа при вызове метода создания/редактирования адреса при ручном вводе. |

* + 1. Удаление площадки, занимаемой организацией

**DELETE api/PassportMO/MOArea**

Удаление площадки, занимаемой организацией

|  |  |
| --- | --- |
| **Входящие параметры** | * MOArea\_id (N, O) – Идентификатор площадки, занимаемой организацией. |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление связи с транспортными узлами

**POST api/PassportMO/TransportConnect**

Добавление связи плащадки с транспортными узлами

|  |  |
| --- | --- |
| **Входящие параметры** | * MOArea\_id (N, O) – Идентификатор площадки, занимаемой организацией. Фильтруется по методу GET api/PassportMO/MOAreaListByMO * TransportConnect\_Airport (T[256], H) – Ближайший аэропорт. * TransportConnect\_DisAirport (N, H) – Расстояние до аэропорта, км. * TransportConnect\_Heliport (T[256], H) – Ближайшая вертолетная площадка. * TransportConnect\_DisHeliport (N, H) - Расстояние до вертолетной площадки, км. * TransportConnect\_Station (T[256], H) – Ближайшая станция. * TransportConne ct\_DisStation (N, H) – Расстояние до станции, км. * TransportConnect\_Railway (T[256], H) – Ближайший автовокзал. * TransportConnect\_DisRailway (N, H) – Расстояние до автовокзала, км. * TransportConnect\_MainRoad (T[256], H) – Ближайшая главная дорога. |
| **Ответ** | **Успешный ответ**   * TransportConnect\_id (N, O) – Идентификатор связи с транспортными узлами. |

* + 1. Изменение связи с транспортными узлами

**PUT api/PassportMO/TransportConnect**

Изменение связи плащадки с транспортными узлами

|  |  |
| --- | --- |
| **Входящие параметры** | * TransportConnect\_id (N, O) – Идентификатор связи с транспортными узлами. * MOArea\_id (N, H) – Идентификатор площадки, занимаемой организацией. Фильтруется по методу GET api/PassportMO /MOAreaListByMO * TransportConnect\_Airport (T[256], H) – Ближайший аэропорт. * TransportConnect\_DisAirport (N, H) – Расстояние до аэропорта, км. * TransportConnect\_Heliport (T[256], H) – Ближайшая вертолетная площадка. * TransportConnect\_DisHeliport (N, H) - Расстояние до вертолетной площадки, км. * TransportConnect\_Station (T[256], H) – Ближайшая станция. * TransportConne ct\_DisStation (N, H) – Расстояние до станции, км. * TransportConnect\_Railway (T[256], H) – Ближайший автовокзал. * TransportConnect\_DisRailway (N, H) – Расстояние до автовокзала, км. * TransportConnect\_MainRoad (T[256], H) – Ближайшая главная дорога. |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0. |

* + 1. Получение списка связей площадки с транспортными узлами

**GET api/PassportMO/TransportConnectListByMOArea**

Получение списка связей площадки с транспортными узлами

|  |  |
| --- | --- |
| **Входящие параметры** | * MOArea\_id (N, O) – Идентификатор площадки |
| **Ответ** | **Успешный ответ** - список:   * TransportConnect\_id (N, O) – Идентификатор связи с транспортными узлами. |

* + 1. Получение атрибутов связи с транспортными узлами по идентификатору.

**GET api/PassportMO/TransportConnectById**

Получение атрибутов связи с транспортными узлами по идентификатору.

|  |  |
| --- | --- |
| **Входящие параметры** | * TransportConnect\_id (N, O) – Идентификатор связи с транспортными узлами. |
| **Ответ** | **Успешный ответ**   * MOArea\_id (N, O) – Идентификатор площадки, занимаемой организацией. * TransportConnect\_Airport (T[256], H) – Ближайший аэропорт. * TransportConnect\_DisAirport (N, H) – Расстояние до аэропорта, км. * TransportConnect\_Heliport (T[256], H) – Ближайшая вертолетная площадка. * TransportConnect\_DisHeliport (N, H) - Расстояние до вертолетной площадки, км. * TransportConnect\_Station (T[256], H) – Ближайшая станция. * TransportConne ct\_DisStation (N, H) – Расстояние до станции, км. * TransportConnect\_Railway (T[256], H) – Ближайший автовокзал. * TransportConnect\_DisRailway (N, H) – Расстояние до автовокзала, км. * TransportConnect\_MainRoad (T[256], H) – Ближайшая главная дорога. |

* + 1. Удаление связи с транспортными узлами

**DELETE api/PassportMO/TransportConnect**

Удаление связи с транспортными узлами

|  |  |
| --- | --- |
| **Входящие параметры** | * TransportConnect\_id (N, O) – Идентификатор связи с транспортными узлами. |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление здания МО

**POST api/PassportMO/LpuBuildingPass**

Добавление здания МО

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. * LpuBuildingPass\_AmbPlace (N, H) – Амбулаторные места * LpuBuildingPass\_BuildingIdent (T[64], O) – Идентификатор здания (Уникальный номер здания по учету учреждения) * LpuBuildingPass\_PowerProjBed (N, H) – Мощность по проекту (число коек) * LpuBuildingPass\_PowerProjViz (N, H) – Мощность по проекту (число посещений) * LpuBuildingPass\_IsBalance (N,О) – Признак на балансе. Значение типа флаг. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * LpuBuildingPass\_Name (T[50], O) – Наименование здания * BuildingAppointmentType\_id (N, H) – Идентификатор назначения здания. Значение из справочника dbo.BuildingAppointmentType * LpuBuildingPass\_BuildVol (N, H) – Объем здания * LpuBuildingPass\_TotalArea (N, H) – Общая площадь * MOArea\_id (N, O) – Идентификатор площадки, занимаемой организацией. Нужно отфильтровать по МО. * LpuBuildingPass\_OfficeArea (N, H) – Площадь кабинетов * LpuBuildingPass\_BedArea (N, H) – Площадь коечных отделений * LpuBuildingPass\_EffBuildVol (N, H) – Полезная площадь * PropertyType\_id (N, H) – Форма владения. Значение из справочника dbo.PropertyType * LpuBuildingPass\_StatPlace (N, H) – Стационарные места * LpuBuildingPass\_MedWorkCabinet (N, H) – Число кабинетов врачебного приема * LpuBuildingType\_id (N,О) – Вид здания по применению. Значение из справочника dbo.LpuBuildingType * LpuBuildingPass\_IsVentil (N, O) – Признак вентиляции. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * LpuBuildingPass\_YearRepair (D, H) – Год последней реконструкции (капитального ремонта) * LpuBuildingPass\_YearBuilt (D, O) – Дата постройки * LpuBuildingPass\_YearProjDoc (D, H) – Дата разработки проектной документации * BuildingTechnology\_id (N, O) – Идентификатор типа по классу технологий. Значение из справочника passport.BuildingTechnology * BuildingHoldConstrType\_id (N, H) – Идентификатор несущей конструкции. Значение из справочника dbo.BuildingHoldConstrType * LpuBuildingPass\_NumProj (T, H) – Номер проекта. * BuildingOverlapType\_id (N, H) – Идентификатор типа перекрытий. Значение из справочника dbo.BuildingOverlapType * BuildingCurrentState\_id (N, H) – Идентификатор текущего состояния здания. Значение из справочника fed.BuildingCurrentState * BuildingType\_id (N, H) – Идентификатор типа проекта здания. Значение из справочника dbo.BuildingType * LpuBuildingPass\_Floors (N, H) – Этажность. * LpuBuildingPass\_IsDomesticGas (N, О) – Признак наличия бытового газоснабжения. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * DHotWater\_id (N, H) – Идентификатор горячего водоснабжения. Значение из справочника fed.DHotWater * DCanalization\_id (N, H) – Идентификатор канализации. Значение из справочника fed.DCanalization * LpuBuildingPass\_IsAirCond (N, О) – Признак наличия кондиционирования. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * LpuBuildingPass\_IsMedGas (N, О) – Признак наличия лечебного газоснабжения. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * DHeating\_id (N, H) – Идентификатор отопления. Значение из справочника fed.DHeating * LpuBuildingPass\_IsColdWater (N, О) – Признак наличия холодного водоснабжения. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * LpuBuildingPass\_HostLift (N, H) – Число медицинских лифтов. * LpuBuildingPass\_PassLift (N, H) – Число пассажирских лифтов. * LpuBuildingPass\_IsElectric (N, О) – Признак наличия электроснабжения. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * FuelType\_id (N, H) – Идентификатор вида топлива отопления. Значение из справочника passport.FuelType * DLink\_id (N, H) – Идентификатор канала связи. Значение из справочника fed.DLink * LpuBuildingPass\_IsFreeEnergy (N, O) – Наличие независимых источников энергоснабжения. Значение типа флаг. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * LpuBuildingPass\_ValDT (D, H) – Дата оценки стоимости. * LpuBuildingPass\_WearPersent (N, H) – Процент износа. * LpuBuildingPass\_ResidualCost (N, H) – Остаточная стоимость. * LpuBuildingPass\_PurchaseCost (N, H) – Первоначальная стоимость. * LpuBuildingPass\_FactVal (N, H) – Фактическая стоимость. * LpuBuildingPass\_IsAutoFFSig (N, O) – Наличие автоматической пожарной сигнализации в здании. Значение типа флаг. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * LpuBuildingPass\_IsFFOutSignal (N, O) – Наличие вывода сигнала о срабатывании систем противопожарной защиты в подразделении пожарной охраны в здании. Значение типа флаг. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * LpuBuildingPass\_IsCallButton (N, O) – Наличие кнопки (брелока) экстренного вызова милиции в здании. Значение типа флаг. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * LpuBuildingPass\_CountDist (N, H) – Количество нарушений требований пожарной безопасности * LpuBuildingPass\_IsEmergExit (N, O) – Наличие эвакуационных путей и выходов в здании. Значение типа флаг. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * LpuBuildingPass\_StretProtect (N, O) – Признак обеспеченности персонала здания учреждения носилками для эвакуации маломобильных пациентов. Значение типа флаг. * LpuBuildingPass\_RespProtect(N, O) – Признак обеспеченности персонала здания учреждения средствами индивидуальной защиты органов дыхания. Значение типа флаг. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * LpuBuildingPass\_IsSecurAlarm (N, O) – Наличие охранной сигнализации в здании. Значение типа флаг. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * LpuBuildingPass\_IsFFWater (N, O) – Наличие противопожарного водоснабжения в здании. Значение типа флаг. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * LpuBuildingPass\_IsConnectFSecure (N, O) – Наличие прямой телефонной связи с подразделением пожарной охраны здания. Значение типа флаг. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * LpuBuildingPass\_IsWarningSys (N, O) – Наличие системы оповещения и управления эвакуацией людей при пожаре в здании. Значение типа флаг. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * LpuBuildingPass\_FSDis (N, H) – Удаление от ближайшего пожарного подразделения. * LpuBuildingPass\_IsBuildEmerg (N, О) – Признак аварийного состояния Возможные значения: 0 и 1, - где 0 – нет, 1 – да * LpuBuildingPass\_IsNeedCap (N, О) – Признак требования капитального ремонта Возможные значения: 0 и 1, - где 0 – нет, 1 – да * LpuBuildingPass\_IsNeedRec (N, О) – Признак требования реконструкции Возможные значения: 0 и 1, - где 0 – нет, 1 – да * LpuBuildingPass\_IsNeedDem (N, О) – Признак требования сноса Возможные значения: 0 и 1, - где 0 – нет, 1 – да |
| **Ответ** | **Успешный ответ**   * LpuBuildingPass\_id (N, O) – Идентификатор здания МО |

* + 1. Изменение атрибутов здания МО

**PUT api/PassportMO/LpuBuildingPass**

Изменение атрибутов здания МО

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuBuildingPass\_id (N, O) – Идентификатор здания МО * LpuBuildingPass\_AmbPlace (N, H) – Амбулаторные места * LpuBuildingPass\_BuildingIdent (T[64], H) – Идентификатор здания (Уникальный номер здания по учету учреждения) * LpuBuildingPass\_PowerProjBed (N, H) – Мощность по проекту (число коек) * LpuBuildingPass\_PowerProjViz (N, H) – Мощность по проекту (число посещений) * LpuBuildingPass\_IsBalance (N, H) – Признак на балансе. Значение типа флаг. * LpuBuildingPass\_Name (T[50], H) – Наименование здания * BuildingAppointmentType\_id (N, H) – Идентификатор назначения здания. Значение из справочника dbo.BuildingAppointmentType * LpuBuildingPass\_BuildVol (N, H) – Объем здания * LpuBuildingPass\_TotalArea (N, H) – Общая площадь * MOArea\_id (N, H) – Идентификатор площадки, занимаемой организацией. Нужно отфильтровать по МО. * LpuBuildingPass\_OfficeArea (N, H) – Площадь кабинетов * LpuBuildingPass\_BedArea (N, H) – Площадь коечных отделений * LpuBuildingPass\_EffBuildVol (N, H) – Полезная площадь * PropertyType\_id (N, H) – Форма владения. Значение из справочника dbo.PropertyType * LpuBuildingPass\_StatPlace (N, H) – Стационарные места * LpuBuildingPass\_MedWorkCabinet (N, H) – Число кабинетов врачебного приема * LpuBuildingType\_id (N, H) – Вид здания по применению. Значение из справочника dbo.LpuBuildingType * LpuBuildingPass\_IsVentil (N, H) – Признак вентиляции. * LpuBuildingPass\_YearRepair (D, H) – Год последней реконструкции (капитального ремонта) * LpuBuildingPass\_YearBuilt (D, H) – Дата постройки * LpuBuildingPass\_YearProjDoc (D, H) – Дата разработки проектной документации * BuildingTechnology\_id (N, H) – Идентификатор типа по классу технологий. Значение из справочника passport.BuildingTechnology * BuildingHoldConstrType\_id (N, H) – Идентификатор несущей конструкции. Значение из справочника dbo.BuildingHoldConstrType * LpuBuildingPass\_NumProj (T, H) – Номер проекта. * BuildingOverlapType\_id (N, H) – Идентификатор типа перекрытий. Значение из справочника dbo.BuildingOverlapType * BuildingCurrentState\_id (N, H) – Идентификатор текущего состояния здания. Значение из справочника fed.BuildingCurrentState * BuildingType\_id (N, H) – Идентификатор типа проект здания. Значение из справочника dbo.BuildingType * LpuBuildingPass\_Floors (N, H) – Этажность. * LpuBuildingPass\_IsDomesticGas (N, H) – Признак наличия бытового газоснабжения. * DHotWater\_id (N, H) – Идентификатор горячего водоснабжения. Значение из справочника fed.DHotWater * DCanalization\_id (N, H) – Идентификатор канализации. Значение из справочника fed.DCanalization * LpuBuildingPass\_IsAirCond (N, H) – Признак наличия кондиционирования. * LpuBuildingPass\_IsMedGas (N, H) – Признак наличия лечебного газоснабжения. * DHeating\_id (N, H) – Идентификатор отопления. Значение из справочника fed.DHeating * LpuBuildingPass\_IsColdWater (N, H) – Признак наличия холодного водоснабжения. * LpuBuildingPass\_HostLift (N, H) – Число медицинских лифтов. * LpuBuildingPass\_PassLift (N, H) – Число пассажирских лифтов. * LpuBuildingPass\_IsElectric (N, H) – Признак наличия электроснабжения. * FuelType\_id (N, H) – Идентификатор вида топлива отопления. Значение из справочника passport.FuelType * DLink\_id (N, H) – Идентификатор канала связи. Значение из справочника fed.DLink * LpuBuildingPass\_IsFreeEnergy (N, H) – Наличие независимых источников энергоснабжения. Значение типа флаг. * LpuBuildingPass\_ValDT (D, H) – Дата оценки стоимости. * LpuBuildingPass\_WearPersent (N, H) – Процент износа. * LpuBuildingPass\_ResidualCost (N, H) – Остаточная стоимость. * LpuBuildingPass\_PurchaseCost (N, H) – Первоначальная стоимость. * LpuBuildingPass\_FactVal (N, H) – Фактическая стоимость. * LpuBuildingPass\_IsAutoFFSig (N, H) – Наличие автоматической пожарной сигнализации в здании. Значение типа флаг. * LpuBuildingPass\_IsFFOutSignal (N, H) – Наличие вывода сигнала о срабатывании систем противопожарной защиты в подразделении пожарной охраны в здании. Значение типа флаг. * LpuBuildingPass\_IsCallButton (N, H) – Наличие кнопки (брелока) экстренного вызова милиции в здании. Значение типа флаг. * LpuBuildingPass\_CountDist (N, H) – Количество нарушений требований пожарной безопасности * LpuBuildingPass\_IsEmergExit (N, H) – Наличие эвакуационных путей и выходов в здании. Значение типа флаг. * LpuBuildingPass\_StretProtect (N, H) – Признак обеспеченности персонала здания учреждения носилками для эвакуации маломобильных пациентов. Значение типа флаг. * LpuBuildingPass\_RespProtect(N, H) – Признак обеспеченности персонала здания учреждения средствами индивидуальной защиты органов дыхания. Значение типа флаг. * LpuBuildingPass\_IsSecurAlarm (N, H) – Наличие охранной сигнализации в здании. Значение типа флаг. * LpuBuildingPass\_IsFFWater (N, H) – Наличие противопожарного водоснабжения в здании. Значение типа флаг. * LpuBuildingPass\_IsConnectFSecure (N, H) – Наличие прямой телефонной связи с подразделением пожарной охраны здания. Значение типа флаг. * LpuBuildingPass\_IsWarningSys (N, H) – Наличие системы оповещения и управления эвакуацией людей при пожаре в здании. Значение типа флаг. * LpuBuildingPass\_FSDis (N, H) – Удаление от ближайшего пожарного подразделения. * LpuBuildingPass\_IsBuildEmerg (N, H) – Признак аварийного состояния * LpuBuildingPass\_IsNeedCap (N, H) – Признак требования капитального ремонта * LpuBuildingPass\_IsNeedRec (N, H) – Признак требования реконструкции * LpuBuildingPass\_IsNeedDem (N, H) – Признак требования сноса |
| **Ответ** | **Успешный ответ**  Ошибка с кодом 0. |

* + 1. Получение списка зданий МО

**GET api/PassportMO/LpuBuildingPassListByMO**

Получение списка зданий МО

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,О) - Идентификатор МО. |
| **Ответ** | **Успешный ответ** - список:   * LpuBuildingPass\_id (N, O) – Идентификатор здания МО |

* + 1. Получение здания МО по номеру учета

**GET api/PassportMO/LpuBuildingPassByBuildingIdent**

Получение здания МО по номеру учета

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuBuildingPass\_BuildingIdent (T[64], O) – Идентификатор здания (Уникальный номер здания по учету учреждения) |
| **Ответ** | **Успешный ответ**:   * LpuBuildingPass\_id (N, O) – Идентификатор здания МО |

* + 1. Получение атрибутов здания МО по идентификатору

**GET api/PassportMO/LpuBuildingPassById**

Получение атрибутов здания МО по идентификатору.

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuBuildingPass\_id (N, O) – Идентификатор здания МО |
| **Ответ** | **Успешный ответ**   * LpuBuildingPass\_AmbPlace (N, H) – Амбулаторные места * LpuBuildingPass\_BuildingIdent (T[64], O) – Идентификатор здания (Уникальный номер здания по учету учреждения) * LpuBuildingPass\_PowerProjBed (N, H) – Мощность по проекту (число коек) * LpuBuildingPass\_PowerProjViz (N, H) – Мощность по проекту (число посещений) * LpuBuildingPass\_IsBalance (N,О) – Признак на балансе. Значение типа флаг. * LpuBuildingPass\_Name (T[50], O) – Наименование здания * BuildingAppointmentType\_id (N, H) – Идентификатор назначения здания. Значение из справочника dbo.BuildingAppointmentType * LpuBuildingPass\_BuildVol (N, H) – Объем здания * LpuBuildingPass\_TotalArea (N, H) – Общая площадь * MOArea\_id (N, O) – Идентификатор площадки, занимаемой организацией. Нужно отфильтровать по МО. * LpuBuildingPass\_OfficeArea (N, H) – Площадь кабинетов * LpuBuildingPass\_BedArea (N, H) – Площадь коечных отделений * LpuBuildingPass\_EffBuildVol (N, H) – Полезная площадь * PropertyType\_id (N, H) – Форма владения. Значение из справочника dbo.PropertyType * LpuBuildingPass\_StatPlace (N, H) – Стационарные места * LpuBuildingPass\_MedWorkCabinet (N, H) – Число кабинетов врачебного приема * LpuBuildingType\_id (N,О) – Вид здания по применению. Значение из справочника dbo.LpuBuildingType * LpuBuildingPass\_IsVentil (N, H) – Признак вентиляции. * LpuBuildingPass\_YearRepair (D, H) – Год последней реконструкции (капитального ремонта) * LpuBuildingPass\_YearBuilt (D, O) – Дата постройки * LpuBuildingPass\_YearProjDoc (D, H) – Дата разработки проектной документации * BuildingTechnology\_id (N, O) – Идентификатор типа по классу технологий. Значение из справочника passport.BuildingTechnology * BuildingHoldConstrType\_id (N, H) – Идентификатор несущей конструкции. Значение из справочника dbo.BuildingHoldConstrType * LpuBuildingPass\_NumProj (T, H) – Номер проекта. * BuildingOverlapType\_id (N, H) – Идентификатор типа перекрытий. Значение из справочника dbo.BuildingOverlapType * BuildingCurrentState\_id (N, H) – Идентификатор текущего состояния здания. Значение из справочника fed.BuildingCurrentState * BuildingType\_id (N, H) – Идентификатор типа проект здания. Значение из справочника dbo.BuildingType * LpuBuildingPass\_Floors (N, H) – Этажность. * LpuBuildingPass\_IsDomesticGas (N, H) – Признак наличия бытового газоснабжения. * DHotWater\_id (N, H) – Идентификатор горячего водоснабжения. Значение из справочника fed.DHotWater * DCanalization\_id (N, H) – Идентификатор канализации. Значение из справочника fed.DCanalization * LpuBuildingPass\_IsAirCond (N, H) – Признак наличия кондиционирования. * LpuBuildingPass\_IsMedGas (N, H) – Признак наличия лечебного газоснабжения. * DHeating\_id (N, H) – Идентификатор отопления. Значение из справочника fed.DHeating * LpuBuildingPass\_IsColdWater (N, H) – Признак наличия холодного водоснабжения. * LpuBuildingPass\_HostLift (N, H) – Число медицинских лифтов. * LpuBuildingPass\_PassLift (N, H) – Число пассажирских лифтов. * LpuBuildingPass\_IsElectric (N, H) – Признак наличия электроснабжения. * FuelType\_id (N, H) – Идентификатор вида топлива отопления. Значение из справочника passport.FuelType * DLink\_id (N, H) – Идентификатор канала связи. Значение из справочника fed.DLink * LpuBuildingPass\_IsFreeEnergy (N, O) – Наличие независимых источников энергоснабжения. Значение типа флаг. * LpuBuildingPass\_ValDT (D, H) – Дата оценки стоимости. * LpuBuildingPass\_WearPersent (N, H) – Процент износа. * LpuBuildingPass\_ResidualCost (N, H) – Остаточная стоимость. * LpuBuildingPass\_PurchaseCost (N, H) – Первоначальная стоимость. * LpuBuildingPass\_FactVal (N, H) – Фактическая стоимость. * LpuBuildingPass\_IsAutoFFSig (N, O) – Наличие автоматической пожарной сигнализации в здании. Значение типа флаг. * LpuBuildingPass\_IsFFOutSignal (N, O) – Наличие вывода сигнала о срабатывании систем противопожарной защиты в подразделении пожарной охраны в здании. Значение типа флаг. * LpuBuildingPass\_IsCallButton (N, O) – Наличие кнопки (брелока) экстренного вызова милиции в здании. Значение типа флаг. * LpuBuildingPass\_CountDist (N, H) – Количество нарушений требований пожарной безопасности * LpuBuildingPass\_IsEmergExit (N, O) – Наличие эвакуационных путей и выходов в здании. Значение типа флаг. * LpuBuildingPass\_StretProtect (N, O) – Признак обеспеченности персонала здания учреждения носилками для эвакуации маломобильных пациентов. Значение типа флаг. * LpuBuildingPass\_RespProtect(N, O) – Признак обеспеченности персонала здания учреждения средствами индивидуальной защиты органов дыхания. Значение типа флаг. * LpuBuildingPass\_IsSecurAlarm (N, O) – Наличие охранной сигнализации в здании. Значение типа флаг. * LpuBuildingPass\_IsFFWater (N, O) – Наличие противопожарного водоснабжения в здании. Значение типа флаг. * LpuBuildingPass\_IsConnectFSecure (N, O) – Наличие прямой телефонной связи с подразделением пожарной охраны здания. Значение типа флаг. * LpuBuildingPass\_IsWarningSys (N, O) – Наличие системы оповещения и управления эвакуацией людей при пожаре в здании. Значение типа флаг. * LpuBuildingPass\_FSDis (N, H) – Удаление от ближайшего пожарного подразделения. * LpuBuildingPass\_IsBuildEmerg (N, H) – Признак аварийного состояния * LpuBuildingPass\_IsNeedCap (N, H) – Признак требования капитального ремонта * LpuBuildingPass\_IsNeedRec (N, H) – Признак требования реконструкции * LpuBuildingPass\_IsNeedDem (N, H) – Признак требования сноса |

* + 1. Удаление здания МО

**DELETE api/PassportMO/LpuBuildingPass**

Удаление здания МО

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuBuildingPass\_id (N, O) – Идентификатор здания МО |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Добавление Домового хозяйства

**POST api/PassportMO/LpuHousehold**

Добавление домового хозяйства

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id – (N[8],О) – МО * LpuHousehold\_Name –(T[255],О) – Наименование * LpuHousehold\_ContactPerson–(T[255],О) – Контактное лицо * LpuHousehold\_ContactPhone–(T[10],О) – Контактный телефон * LpuHousehold\_CadNumber–(T[20],Н) – Кадастровый номер * LpuHousehold\_CoordLat–(N[],О) – Координаты (широта) * LpuHousehold\_CoordLon–(N[],О) – Координаты (долгота) * PAddress\_id– (N[8],О) –Адрес хозяйства фактический |
| **Выполняемые действия** | Если идентификатор МО не найден, то ошибка. Данные не создаются.  Иначе: создается домовое хозяйство (fed.LpuHousehold) |
| **Ответ** | **Успешный ответ**   * LpuHousehold\_id – (N[8],Н) –Идентификатор |
| **Пример** | Запрос  Ответ |

* + 1. Изменение Домового хозяйства

**PUT api/PassportMO/** **LpuHousehold**

Изменение домового хозяйства

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuHousehold\_id – (N[8],Н) –Идентификатор * LpuHousehold\_Name –(T[255],О) – Наименование * LpuHousehold\_ContactPerson–(T[255],О) – Контактное лицо * LpuHousehold\_ContactPhone–(T[10],О) – Контактный телефон * LpuHousehold\_CadNumber–(T[20],Н) – Кадастровый номер * LpuHousehold\_CoordLat–(N[],О) – Координаты (широта) * LpuHousehold\_CoordLon–(N[],О) – Координаты (долгота) * PAddress\_id– (N[8],О) –Адрес хозяйства фактический |
| **Выполняемые действия** | Если идентификатор домового хозяйства не найден, то ошибка. Данные не изменяются.  Иначе: изменяются данные домового хозяйства (fed.LpuHousehold) |
| **Ответ** | **При наличии ошибок – код ошики.**  **Успешный ответ - код 0.** |
| **Пример** | Запрос  Ответ |

* + 1. Удаление Домового хозяйства

**DELETE api/PassportMO/** **LpuHousehold**

Удаление домового хозяйства

|  |  |
| --- | --- |
| **Входящие параметры** | * LpuHousehold\_id – (N[8],Н) –Идентификатор |
| **Выполняемые действия** | Если идентификатор домового хозяйства не найден, то ошибка. Данные не изменяются.  Иначе: удаляется данные домового хозяйства (fed.LpuHousehold) |
| **Ответ** | **При наличии ошибок – код ошики.**  **Успешный ответ - код 0.** |
| **Пример** | Запрос  Ответ |

* + 1. Получение списка Домовых хозяйств МО

**GET api/PassportMO/** **LpuHouseholdByMO**

Получение списка Домовых хозяйств МО

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id – (N[8],Н) – Идентификатор МО |
| **Выполняемые действия** | Если идентификатор МО не найден, то ошибка. Данные не изменяются.  Иначе: возвращается список домоых хозяйств МО (fed.LpuHousehold) |
| **Исходящие параметры** | * LpuHousehold\_id – (N[8],Н) –Идентификатор * LpuHousehold\_Name –(T[255],О) – Наименование * LpuHousehold\_ContactPerson–(T[255],О) – Контактное лицо * LpuHousehold\_ContactPhone–(T[10],О) – Контактный телефон * LpuHousehold\_CadNumber–(T[20],Н) – Кадастровый номер * LpuHousehold\_CoordLat–(N[],О) – Координаты (широта) * LpuHousehold\_CoordLon–(N[],О) – Координаты (долгота) * PAddress\_id– (N[8],О) –Адрес хозяйства фактический**.** |
| **Пример** | Запрос  Ответ |

* 1. Описание методов работы с медицинскими изделиями

При редактировании сущностей (методы PUT ) предусмотрено следующее:

* Для случаев изменения параметра, являющегося ссылкой на другую сущность (родительскую) необходимо проверять существование экземпляра сущности с передаваемыми параметрами. Если параметр найден, то обновлять данные. Если параметр не найден, то ошибка.
* Для случаев изменения параметра, являющегося необязательным свойством при создании сущности, перезаписывать свойство (в том числе при получении пустых или нулевых значений).
  1. 1. Получение Медицинского изделия по идентификатору

**GET api/MedProductCard**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedProductCard\_id(N,У) – идентификатор мед изделия * AccountingData\_InventNumber (T,У) – инвентарный номер   Обязателен хотя бы один из парметров |
| **Ответ** | * MedProductCard\_id(N,У) – идентификатор мед изделия * MedProductClass\_id (N,O) - Класс МИ; * AccountingData\_InventNumber (T,O) – инвентарный номер * MedProductCard\_SerialNumber (Т,O) – серийный номер; * AccountingData\_RegNumber (T, H) – Регистрационный знак для автомобилей; * LpuBuilding\_id (N,O) – идентификатор Подразделения; * LpuUnit\_id (N,H) – идентификатор Группы отделения; * LpuSection\_id (N,H) – идентификатор отделения * Org\_id (N,O) – идентификатор поставщика; * MedProductCard\_begDate (D,O) – дата выпуска; * MedProductCard\_UsePeriod (N, H) – Срок использования; * MedProductCard\_IsEducatAct (N,О) – Признак «Наличе акта об обучении мед. персонала».Возможные занчения: 0 и 1; * MedProductCard\_IsNoAvailLpu (N, О) – Признак «Недоступна для МО». Возможные занчения: 0 и 1. ; * RegCertificate\_endDate (D,H) –Срок действия рег. удостоверения; * RegCertificate\_setDate(D,H) - Дата регистрационного удостоверения; * RegCertificate\_Number (T, Н) - Номер регистрационного удостоверения; * RegCertificate\_OrderNumber (T, Н) – Номер приказа; * RegCertificate\_MedProductName (T,H) -Наименование МИ по регистрационным документам; * Org\_regid (N,H) – Идентификатор организации держателя удостоверения; * Org\_prid (N,H) –Идентификатор организации производителя; * Org\_decid (N, H) – Идентификатор организации декларанта; * MedProductCard\_Options (T,H) – комплектация; * MedProductCard\_OtherParam (T, H) – прочие параметры; * MeasureFund\_IsMeasure (N,О) –признак «Является средством измерения». Возможные значения: 0 и 1; * MeasureFund\_Range (T, Н) – Диапазон измерений; * OkeiLink\_id (N,У) –Ед. Измерения (справочник ОКЕИ). Обязательно если MeasureFund\_IsMeasure=1; * MeasureFund\_RegNumbe (T, H) - Регистрационный номер средств измерения; * MeasureFund\_AccuracyClass (T, Н) – Класс точности средств измерения; * AccountingData\_buyDate (D,O) – дата приобретения; * AccountingData\_begDate (D,H) – дата принятия на учет; * GosContract\_Number (T, H) – номер гос контракта; * AccountingData\_setDate(D,H) – дата ввода в эксплуатацию; * AccountingData\_endDate(D, H) –вывод из эксплуатации * FinancingType\_id (N,O) – идентификатор справочника программа закупки: Cправочник «Приобретено в рамках» passport.FinancingType * AccountingData\_ProductCost (N,O) – Цена производителя; * PropertyType\_id(N,O) – Форма владения; Справочник «Форма владения» dbo.PropertyType; * GosContract\_setDate (D,H) – Дата заключения контракта; * AccountingData\_BuyCost (N,O) – стоимость приобретения; * DeliveryType\_id (N,O) – тип поставки; Пополняемый справочник passport.DeliveryType; * MedProductCard\_DocumentTO (Т,H) - Документ, подтверждающий прохождение ТО; * MedProductCard\_IsContractTO (N,O) Наличие договора на тех. обслуживание. Возможные значения : 0 и 1. * Org\_toid(N,У) – Идентификатор организации, осуществляющей тех. обслуживание. Обязателен если MedProductCard\_IsContractTO=1; * MedProductCard\_IsOrgLic(N,O) - Наличие лицензии на проведение тех. обслуживания. Возможные значения : 0 и 1.. * MedProductCard\_IsLpuLic(N,О) - Наличие лицензии у МО на проведение тех. обслуживания. Возможные значения : 0 и 1. * MedProductCard\_IsRepair (N,O) – признак «Требует ремонта». Возможные значения : 0 и 1.; * MedProductCard\_IsSpisan (N,O) – признак «Требует ремонта» Возможные значения : 0 и 1; * MedProductCard\_RepairDate(D,У) – Дата установки статуса: 'Требуетремонта'. Обязателен если MedProductCard\_IsRepair=1; * MedProductCard\_SpisanDate(D,У) - Дата установки статуса: 'Требуетсписания'. Обязателен если MedProductCard\_IsSpisan=1 * MedProductCard\_SetResource (N,H) - Установленный/назначенный ресурс (ед.). Возможные значения: числа с плавающей точкой (float). * MedProductCard\_AvgProcTime(N,H) - Средняя длительность процедуры (ед.). Возможные значения: числа с плавающей точкой (float). |

* + 1. Создание Медицинского изделия

**POST api/MedProductCard**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedProductClass\_id (N,O) - Класс МИ; * AccountingData\_InventNumber (T,O) – инвентарный номер. Значение уникальное * MedProductCard\_SerialNumber (Т,O) – серийный номер; * AccountingData\_RegNumber (T, H) – Регистрационный знак для автомобилей; * LpuBuilding\_id (N,O) – идентификатор Подразделения; * LpuUnit\_id (N,H) – идентификатор Группы отделения; * LpuSection\_id (N,H) – идентификатор отделения * Org\_id (N,O) – идентификатор поставщика; * MedProductCard\_begDate (D,O) – дата выпуска; * MedProductCard\_UsePeriod (N, H) – Срок использования; * MedProductCard\_IsEducatAct (N,О) – Признак «Наличе акта об обучении мед. персонала». Возможные значения: 0 и 1, - где 0 – нет, 1 – да. * MedProductCard\_IsNoAvailLpu (N, О) – Признак «Недоступна для МО». Возможные значения: 0 и 1, - где 0 – нет, 1 – да.Если не передается, то присваивать 0; * RegCertificate\_endDate (D,H) –Срок действия рег. удостоверения; * RegCertificate\_setDate(D,H) - Дата регистрационного удостоверения; * RegCertificate\_Number (T, Н) - Номер регистрационного удостоверения; * RegCertificate\_OrderNumber (T, Н) – Номер приказа; * RegCertificate\_MedProductName (T,H) -Наименование МИ по регистрационным документам; * Org\_regid (N,H) – Идентификатор организации держателя удостоверения; * Org\_prid (N,H) –Идентификатор организации производителя; * Org\_decid (N, H) – Идентификатор организации декларанта; * MedProductCard\_Options (T,H) – комплектация; * MedProductCard\_OtherParam (T, H) – прочие параметры; * MeasureFund\_IsMeasure (N,О) –признак «Является средством измерения». Возможные значения: 0 и 1, - где 0 – нет, 1 – да.; * MeasureFund\_Range (T, Н) – Диапазон измерений; * OkeiLink\_id (N,У) –Ед. Измерения (справочник ОКЕИ). Обязательно если MeasureFund\_IsMeasure=1; * MeasureFund\_RegNumber (T, H) - Регистрационный номер средств измерения; * MeasureFund\_AccuracyClass (T, Н) – Класс точности средств измерения; * AccountingData\_buyDate (D,O) – дата приобретения; * AccountingData\_begDate (D,H) – дата принятия на учет; * GosContract\_Number (T, H) – номер гос контракта; * AccountingData\_setDate(D,H) – дата ввода в эксплуатацию; * AccountingData\_endDate(D, H) –вывод из эксплуатации * FinancingType\_id (N,O) – идентификатор справочника программа закупки: Cправочник «Приобретено в рамках» passport.FinancingType * AccountingData\_ProductCost (N,O) – Цена производителя; * PropertyType\_id(N,O) – Форма владения; Справочник «Форма владения» dbo.PropertyType; * GosContract\_setDate (D,H) – Дата заключения контракта; * AccountingData\_BuyCost (N,O) – стоимость приобретения; * DeliveryType\_id (N,O) – тип поставки; Пополняемый справочник passport.DeliveryType; * MedProductCard\_DocumentTO (Т,H) - Документ, подтверждающий прохождение ТО;   MedProductCard\_IsContractTO (N,O) Наличие договора на тех. обслуживание. Возможные значения: 0 и 1, - где 0 – нет, 1 – да..   * Org\_toid(N,У) – Идентификатор организации, осуществляющей тех. обслуживание. Обязателен если MedProductCard\_IsContractTO=1; * MedProductCard\_IsOrgLic(N,O) - Наличие лицензии на проведение тех. обслуживания. Возможные значения: 0 и 1, - где 0 – нет, 1 – да. * MedProductCard\_IsLpuLic(N,О) - Наличие лицензии у МО на проведение тех. обслуживания. Возможные значения: 0 и 1, - где 0 – нет, 1 – да. * MedProductCard\_IsRepair (N,O) – признак «Требует ремонта». Возможные значения: 0 и 1, - где 0 – нет, 1 – да; * MedProductCard\_IsSpisan (N,O) – признак «Требует ремонта». Возможные значения: 0 и 1, - где 0 – нет, 1 – да.; * MedProductCard\_RepairDate(D,У) – Дата установки статуса: 'Требуетремонта'. Обязателен если MedProductCard\_IsRepair=1; * MedProductCard\_SpisanDate(D,У) - Дата установки статуса: 'Требуетсписания'. Обязателен если MedProductCard\_IsSpisan=1. * MedProductCard\_SetResource (N,H) - Установленный/назначенный ресурс (ед.). Возможные значения: числа с плавающей точкой (float). * MedProductCard\_AvgProcTime(N,H) - Средняя длительность процедуры (ед.). Возможные значения: числа с плавающей точкой (float). |
| **Ответ** | MedProductCard\_id(N,O) – идентификатор мед изделия |

* + 1. Редактирование Медицинского изделия

**PUT api/MedProductCard**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedProductCard\_id (N,O) - идентификатор мед изделия * MedProductClass\_id (N,Н) - Класс МИ; * AccountingData\_InventNumber (T,Н) – инвентарный номер. Значение уникальное * MedProductCard\_SerialNumber (Т,Н) – серийный номер; * AccountingData\_RegNumber (T, H) – Регистрационный знак для автомобилей; * LpuBuilding\_id (N,Н) – идентификатор Подразделения; * LpuUnit\_id (N,H) – идентификатор Группы отделения; * LpuSection\_id (N,H) – идентификатор отделения * Org\_id (N,Н) – идентификатор поставщика; * MedProductCard\_begDate (D,Н) – дата выпуска; * MedProductCard\_UsePeriod (N, H) – Срок использования; * MedProductCard\_IsEducatAct (N,H) – Признак «Наличе акта об обучении мед. персонала»; * MedProductCard\_IsNoAvailLpu (N, H) – Признак «Недоступна для МО»; * RegCertificate\_endDate (D,H) –Срок действия рег. удостоверения; * RegCertificate\_setDate(D,H) - Дата регистрационного удостоверения; * RegCertificate\_Number (T, Н) - Номер регистрационного удостоверения; * RegCertificate\_OrderNumber (T, Н) – Номер приказа; * RegCertificate\_MedProductName (T,H) -Наименование МИ по регистрационным документам; * Org\_regid (N,H) – Идентификатор организации держателя удостоверения; * Org\_prid (N,H) –Идентификатор организации производителя; * Org\_decid (N, H) – Идентификатор организации декларанта; * MedProductCard\_Options (T,H) – комплектация; * MedProductCard\_OtherParam (T, H) – прочие параметры; * MeasureFund\_IsMeasure (N,H) –признак «Является средством измерения»; * MeasureFund\_Range (T, Н) – Диапазон измерений; * OkeiLink\_id (N,Н) –Ед. Измерения (справочник ОКЕИ). Обязательно если MeasureFund\_IsMeasure=1; * MeasureFund\_RegNumber (T, H) - Регистрационный номер средств измерения; * MeasureFund\_AccuracyClass (T, Н) – Класс точности средств измерения; * AccountingData\_buyDate (D,Н) – дата приобретения; * AccountingData\_begDate (D,H) – дата принятия на учет; * GosContract\_Number (T, H) – номер гос контракта; * AccountingData\_setDate(D,H) – дата ввода в эксплуатацию; * AccountingData\_endDate(D, H) –вывод из эксплуатации * FinancingType\_id (N,Н) – идентификатор справочника программа закупки: Cправочник «Приобретено в рамках» passport.FinancingType * AccountingData\_ProductCost (N,Н) – Цена производителя; * PropertyType\_id(N,Н) – Форма владения; Справочник «Форма владения» dbo.PropertyType; * GosContract\_setDate (D,H) – Дата заключения контракта; * AccountingData\_BuyCost (N,Н) – стоимость приобретения; * DeliveryType\_id (N,Н) – тип поставки; Пополняемый справочник passport.DeliveryType; * MedProductCard\_DocumentTO (Т,H) - Документ, подтверждающий прохождение ТО; * MedProductCard\_IsContractTO (N,O) Наличие договора на тех. обслуживание. Возможные значения : 0 и 1. * Org\_toid(N,У) – Идентификатор организации, осуществляющей тех. обслуживание. Обязателен если MedProductCard\_IsContractTO=1; * MedProductCard\_IsOrgLic(N,O) - Наличие лицензии на проведение тех. обслуживания. Возможные значения : 0 и 1.. * MedProductCard\_IsLpuLic(N,О) - Наличие лицензии у МО на проведение тех. обслуживания. Возможные значения : 0 и 1. * MedProductCard\_IsRepair (N,O) – признак «Требует ремонта». Возможные значения : 0 и 1.; * MedProductCard\_IsSpisan (N,O) – признак «Требует ремонта» Возможные значения : 0 и 1; * MedProductCard\_RepairDate(D,У) – Дата установки статуса: 'Требуетремонта'. Обязателен если MedProductCard\_IsRepair=1; * MedProductCard\_SpisanDate(D,У) - Дата установки статуса: 'Требуетсписания'. Обязателен если MedProductCard\_IsSpisan=1 * MedProductCard\_SetResource (N,H) - Установленный/назначенный ресурс (ед.). Возможные значения: числа с плавающей точкой (float). * MedProductCard\_AvgProcTime(N,H) - Средняя длительность процедуры (ед.). Возможные значения: числа с плавающей точкой (float). |
| **Ответ** | MedProductCard\_id (N,O) – идентификатор мед изделия |

* + 1. Получение списка Медицинских изделий одного Класса

**GET api/MedProductCardListByMedProductClassId**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedProductClass\_id (N,О) – идентификатор записи «Классификатор Медицинского изделия»; |
| **Ответ** | Необходимо выбрать все записи MedProductCard у которых параметр MedProductClass\_id соответствует входящему параметру запроса.  В ответе необходимо возвращать список со следующими параметрами:   * MedProductCard\_id(N,О) – идентификатор мед изделия |

* + 1. Подсчет количества Медицинских изделий одного Класса

**GET api/MedProductCardCountByMedProductClassId**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedProductClass\_id (N,О) – идентификатор записи «Классификатор Медицинского изделия»; |
| **Ответ** | Необходимо выбрать все записи MedProductCard у которых параметр MedProductClass\_id соответствует входящему параметру запроса.  В ответе необходимо возвращать количество найденных записей. |

* + 1. Получение «Классификатор Медицинского изделия»

**GET api/MedProductClass**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedProductClass\_id (N,У) – идентификатор записи «Классификатор Медицинского изделия»; Обязаетлен, если не задан ни один из условно обязательных параметров. * MedProductClass\_Name (T,У) – наименование МИ; Передается в паре с MedProductClass\_Model * MedProductClass\_Model(T,У) – Модель МИ. Передается в паре с MedProductClass\_Name; * CardType\_id(N,Н) – Идентификатор справочника «Тип МИ»; * ClassRiskType\_id(N,Н) – Идентификатор справочника «Класс потенциального риска применения»; * FuncPurpType\_id (N,Н) – Идентификатор справочника «Функциональное назначение»; * UseAreaType\_id(N,Н) – идентификатор «Область применения»; * UseSphereType\_id (N,Н) – идентификатор «Сфера применения»; |
| **Ответ** | * MedProductClass\_id (N,У) – идентификатор записи «Классификатор Медицинского изделия»; * MedProductClass\_Name (T,O) – наименование МИ; * MedProductClass\_Model(T,O) – Модель МИ; * MedProductType\_id (N,H) – идентификатор справочиника «Вид МИ» * CardType\_id(N,O) – Идентификатор справочника «Тип МИ»; * ClassRiskType\_id(N,O) – Идентификатор справочника «Класс потенциального риска применения»; * FuncPurpType\_id (N,O) – Идентификатор справочника «Функциональное назначение»; * UseAreaType\_id(N,O) – идентификатор «Область применения»; * UseSphereType\_id (N,O) – идентификатор «Сфера применения»; * FZ30Type\_id (N,H) – идентификатор «30й ФЗ»; * TNDEDType\_id (N,H) – идентификатор «ТН ВЭД»; * GMDNType\_id (N,H) – идентификатор справочника GMDN; * MT97Type\_id (N,H) – идентификатор МТ по 97пр; * OKOFType\_id (N,H) – идентификатор ОКОФ; * OKPType\_id (N,H) – идентификатор OKP; * OKPDType\_id (N,H) –идентификатор ОКПД; * MedProductClass\_IsAmbulNovor (N,H) – признак «Ренимобиль для новорожденных и детей раннего возраста; * MedProductClass\_IsAmbulTerr (N,H) –признак «Реанимобиль повышенной необходимости. |

* + 1. Создание «Классификатор Медицинского изделия»

**POST api/MedProductClass**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedProductClass\_Name (T,O) – наименование МИ; * MedProductClass\_Model(T,O) – Модель МИ; * MedProductType\_id (N,H) – идентификатор справочиника «Вид МИ» * CardType\_id(N,O) – Идентификатор справочника «Тип МИ»; * ClassRiskType\_id(N,O) – Идентификатор справочника «Класс потенциального риска применения»; * FuncPurpType\_id (N,O) – Идентификатор справочника «Функциональное назначение»; * UseAreaType\_id(N,O) – идентификатор «Область применения»; * UseSphereType\_id (N,O) – идентификатор «Сфера применения»; * FZ30Type\_id (N,H) – идентификатор «30й ФЗ»; * TNDEDType\_id (N,H) – идентификатор «ТН ВЭД»; * GMDNType\_id (N,H) – идентификатор справочника GMDN; * MT97Type\_id (N,H) – идентификатор МТ по 97пр; * OKOFType\_id (N,H) – идентификатор ОКОФ; * OKPType\_id (N,H) – идентификатор OKP; * OKPDType\_id (N,H) –идентификатор ОКПД; * MedProductClass\_IsAmbulNovor (N,H) – признак «Ренимобиль для новорожденных и детей раннего возраста. Принимает значения 0 и 1, где 0 – нет, 1 -да; * MedProductClass\_IsAmbulTerr (N,H) –признак «Реанимобиль повышенной необходимости. Принимает значения 0 и 1, где 0 – нет, 1 –да. |
| **Ответ** | MedProductClass\_id (N,O) – идентификатор |

* + 1. Редактирование «Классификатор Медицинского изделия»

**PUT api/MedProductClass**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedProductClass\_id (N,Н) – идентификатор * MedProductClass\_Name (T,Н) – наименование МИ; * MedProductClass\_Model(T,Н) – Модель МИ; * MedProductType\_id (N,H) – идентификатор справочиника «Вид МИ» * CardType\_id(N,Н) – Идентификатор справочника «Тип МИ»; * ClassRiskType\_id(N,Н) – Идентификатор справочника «Класс потенциального риска применения»; * FuncPurpType\_id (N,Н) – Идентификатор справочника «Функциональное назначение»; * UseAreaType\_id(N,Н) – идентификатор «Область применения»; * UseSphereType\_id (N,Н) – идентификатор «Сфера применения»; * FZ30Type\_id (N,H) – идентификатор «30й ФЗ»; * TNDEDType\_id (N,H) – идентификатор «ТН ВЭД»; * GMDNType\_id (N,H) – идентификатор справочника GMDN; * MT97Type\_id (N,H) – идентификатор МТ по 97пр; * OKOFType\_id (N,H) – идентификатор ОКОФ; * OKPType\_id (N,H) – идентификатор OKP; * OKPDType\_id (N,H) –идентификатор ОКПД; * MedProductClass\_IsAmbulNovor (N,H) – признак «Ренимобиль для новорожденных и детей раннего возраста. Принимает значения 0 и 1, где 0 – нет, 1 -да; * MedProductClass\_IsAmbulTerr (N,H) –признак «Реанимобиль повышенной необходимости. Принимает значения 0 и 1, где 0 – нет, 1 –да. |
| **Ответ** | Если для выбранного классификатора МИ имеются созданные экземпляры МИ (карточки МИ), то вернуть ошибку «Редактирование класса МИ не доступно». |

Если для выбранного классификатора МИ имеются созданные экземпляры МИ (карточки МИ), редактирование класса МИ не доступно.

* + 1. Получение записи «Расходные материалы» по идентификатору

**GET api/MedProductCard /ConsumablesById**

|  |  |
| --- | --- |
| **Входящие параметры** | * Consumables\_id (N,O) – идентификатор записи «Расходные материалы |
| **Ответ** | * MedProductCard\_id(N,O) – идентификатор мед изделия * Consumables\_Name (T,O) – наименование расходного материала» |

* + 1. Получение записи «Расходные материалы» по наименованию и мед изделию

**GET api/MedProductCard /ConsumablesByPar**

|  |  |
| --- | --- |
| **Входящие параметры** | * Consumables\_Name (T,O) – наименование рас ходного материала» * MedProductCard\_id(N,O) – идентификатор мед изделия |
| **Ответ** | * Consumables\_id (N,O) – идентификатор записи «Расходные материалы» * Consumables\_Name (T,O) – наименование рас ходного материала» |

* + 1. Создание записи «Расходные материалы»

**POST api/MedProductCard /Consumables**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedProductCard\_id(N,O) – идентификатор мед изделия * Consumables\_Name (T,O) – наименование расходного материала. Для одного МИ значение должно быть уникальным; |
| **Ответ** | Consumables\_id (N,O) – идентификатор записи «Расходные материалы» |

* + 1. Редактирование записи «Расходные материалы»

**PUT api/MedProductCard /Consumables**

|  |  |
| --- | --- |
| **Входящие параметры** | * Consumables\_id (N,O) – идентификатор записи «Расходные материалы» * MedProductCard\_id(N,Н) – идентификатор мед изделия * Consumables\_Name (T,Н) – наименование расходного материала; |

* + 1. Получение записи «Свидетельство о проверке»

**GET api/MedProductCard/MeasureFundCheck**

|  |  |
| --- | --- |
| Входящие параметры | * MeasureFundCheck\_id(N,У) – идентификатор записи Свидетельства о проверке. Обязателен, если ни один другой параметр не указан * MedProductCard\_id(N,У) – идентификатор мед изделия. Обязательно в группе с MeasureFundCheck\_endDate и MeasureFundCheck\_ Number * MeasureFundCheck\_Number(Т,У) – Номер свидетельства о проверки; Обязательно в группе с MedProductCard\_id и MeasureFundCheck\_endDate. * MeasureFundCheck\_endDate (D,У) – Срок действия свидетельства. Обязателен в группе с MeasureFundCheck\_Number и MedProductCard\_id |
| **Ответ** | * MeasureFundCheck\_id(N,O) – идентификатор записи Свидетельства о проверке * MeasureFundCheck\_Number(Т,О) – Номер свидетельства о проверки; * MeasureFundCheck\_setDate (D, О) –Дата свидетельства о проверке; Уникальное значение для одного МИ * MeasureFundCheck\_endDate (D,О) – Срок действия свидетельства |

* + 1. Создание записи «Свидетельство о проверке»

**POST api/MedProductCard /MeasureFundCheck**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedProductCard\_id(N,O) – идентификатор мед изделия * MeasureFundCheck\_Number(Т,О) – Номер свидетельства о проверке; * MeasureFundCheck\_setDate (D, О) –Дата свидетельства о проверке. Для МИ поле уникальное; * MeasureFundCheck\_endDate (D,О) – Срок действия свидетельства |
| **Ответ** | MeasureFundCheck\_id(N,O) – идентификатор записи Свидетельства о проверке |

Созданеие Свидетельства о проверке доступно если у МИ отмечен признак MeasureFund\_IsMeasure.

* + 1. Редактирование записи «Свидетельство о проверке»

**PUT api/MedProductCard /MeasureFundCheck**

|  |  |
| --- | --- |
| **Входящие параметры** | * MeasureFundCheck\_id(N,O) – идентификатор записи Свидетельства о проверке * MedProductCard\_id(N,H) – идентификатор мед изделия * MeasureFundCheck\_Number(Т,H) – Номер свидетельства о проверки; * MeasureFundCheck\_setDate (D, H) –Дата свидетельства о проверке; * MeasureFundCheck\_endDate (D,H) – Срок действия свидетельства |

* + 1. Получение записи «Начисление износа»

**GET api/MedProductCard /Amortization**

|  |  |
| --- | --- |
| **Входящие параметры** | * Amortization\_id (N,У) – идентичикатор записи «Начисление износа». Обязателен, если не указан ни один другой параметр; * MedProductCard\_id(N,У) – идентификатор мед изделия. Параметр обязателен в паре с Amortization\_setDate; * Amortization\_setDate (D,У) –дата оценки; Параметр обязателен в паре MedProductCard\_id |
| **Ответ** | * Amortization\_id (N,O) – идентичикатор записи «Начисление износа» * MedProductCard\_id(N,O) – идентификатор мед изделия * Amortization\_setDate (D,O) –дата оценки; Уникальная для МИ * Amortization\_FactCost (N, О) –фактическая стоимость; * Amortization\_WearPercent (N, О) – процент износа; * Amortization\_ResidCost (N, О) – остаточная стоимость |

* + 1. Создание записи «Начисление износа»

**POST api/MedProductCard /Amortization**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedProductCard\_id(N,O) – идентификатор мед изделия * Amortization\_setDate (D,O) –дата оценки; Уникальная для МИ * Amortization\_FactCost (N, О) –фактическая стоимость; * Amortization\_WearPercent (N, О) – процент износа; * Amortization\_ResidCost (N, О) – остаточная стоимость |
| **Ответ** | Amortization\_id (N,O) – идентичикатор записи «Начисление износа» |

* + 1. Редактирование записи «Начисление износа»

**PUT api/MedProductCard /Amortization**

|  |  |
| --- | --- |
| **Входящие параметры** | * Amortization\_id (N,O) – идентичикатор записи «Начисление износа» * MedProductCard\_id(N,H) – идентификатор мед изделия * Amortization\_setDate (D,H) –дата оценки; Уникальная для МИ * Amortization\_FactCost (N, H) –фактическая стоимость; * Amortization\_WearPercent (N, H) – процент износа; * Amortization\_ResidCost (N, H) – остаточная стоимость |
| **Ответ** | Amortization\_id (N,O) – идентичикатор записи «Начисление износа» |

* + 1. Получение записи «Простой МИ»

**GET api/MedProductCard /DowntimeByPar**

|  |  |
| --- | --- |
| **Входящие параметры** | * Downtime\_id(N,У) – идентификатор записи «Простой МИ». Обязателен, если не передаются другие параметры; * MedProductCard\_id(N,У) – идентификатор мед изделия. Обязателен в группе с Downtime\_begDate; DowntimeCause\_id * Downtime\_begDate(D,У) –дата начала простоя. Обязателен в группе с MedProductCard\_id; DowntimeCause\_id; * DowntimeCause\_id(N,У) – идентификатор причин простоя. Обязателен в группе с MedProductCard\_id и Downtime\_begDate |
| **Ответ** | * Downtime\_id(N,У) – идентификатор записи «Простой МИ». Обязателен, если не передаются другие параметры; * MedProductCard\_id(N,O) – идентификатор мед изделия * Downtime\_begDate(D,O) –дата начала простоя; * Downtime\_endDate(D,O) – дата возобновления работы; * DowntimeCause\_id(N,O) – идентификатор причин простоя. |

* + 1. Получение списка записей «Простой МИ» для МИ

**GET api/MedProductCard /Downtime**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedProductCard\_id(N,O) – идентификатор мед изделия |
| **Ответ** | Список записей, каждая запись которого включает следующие поля   * Downtime\_id(N,O) – идентификатор записи «Простой МИ» * MedProductCard\_id(N,O) – идентификатор мед изделия * Downtime\_begDate(D,O) –дата начала простоя; * Downtime\_endDate(D,O) – дата возобновления работы; * DowntimeCause\_id(N,O) – идентификатор причин простоя. |

* + 1. Создание записи «Простой МИ»

**POST api/MedProductCard /Downtime**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedProductCard\_id(N,O) – идентификатор мед изделия * Downtime\_begDate(D,O) –дата начала простоя; * Downtime\_endDate(D,O) – дата возобновления работы; * DowntimeCause\_id(N,O) – идентификатор причин простоя. |
| **Ответ** | Downtime\_id(N,O) – идентификатор записи «Простой МИ» |

* + 1. Редактирование записи «Простой МИ»

**PUT api/MedProductCard /Downtime**

|  |  |
| --- | --- |
| **Входящие параметры** | * Downtime\_id(N,O) – идентификатор записи «Простой МИ» * MedProductCard\_id(N,H) – идентификатор мед изделия * Downtime\_begDate(D,H) –дата начала простоя; * Downtime\_endDate(D,H) – дата возобновления работы; * DowntimeCause\_id(N,H) – идентификатор причин простоя. |

* + 1. Получение записи «Эксплуатационные данные»

**GET api/MedProductCard /WorkData**

|  |  |
| --- | --- |
| **Входящие параметры** | * WorkData\_id (N,У) – идентификтаор записи «Эксплуатационные данные». Обязательно, если не указан ни один другой параметр; * MedProductCard\_id(N,У) – идентификатор мед изделия. Обязателен в паре с WorkData\_WorkPeriod; * WorkData\_WorkPeriod(D,У) – Период эксплуатации; Дата уникальна для МИ.Первое число месяца.Обязателен в паре с MedProductCard\_id. |
| **Ответ** | * WorkData\_id (N,О) – идентификтаор записи «Эксплуатационные данные». * MedProductCard\_id(N,O) – идентификатор мед изделия * WorkData\_WorkPeriod(D,O) – Период эксплуатации; Дата уникальна для МИ.Первое число месяца. * WorkData\_DayChange(N; О) – Количество смен в сутки; * WorkData\_CountUse(N; О) – Общее количество применений за период; * WorkData\_KolDay(N; О) – Количество рабочих дней в периоде; * WorkData\_AvgUse (N, О) – Среднее количество применений в смену. |

* + 1. Создание записи «Эксплуатационные данные»

**POST api/MedProductCard /WorkData**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedProductCard\_id(N,O) – идентификатор мед изделия * WorkData\_WorkPeriod(D,O) – Период эксплуатации; Дата уникальна для МИ.Первое число месяца. * WorkData\_DayChange(N; О) – Количество смен в сутки. Используется тип данных float; * WorkData\_CountUse(N; О) – Общее количество применений за период; * WorkData\_KolDay(N; О) – Количество рабочих дней в периоде. Используется тип данных float; |
| **Ответ** | * WorkData\_id (N,O) – идентификтаор записи «Эксплуатационные данные» * WorkData\_AvgUse (N, О) – Среднее количество применений в смену. (расчетное значение). Расчитывается по следующей формуле: WorkData\_AvgUse = WorkData\_CountUse/( WorkData\_DayChange\*WorkData\_KolDay) Округление до двух цифр после запятой. |

* + 1. Редактирование записи «Эксплуатационные данные»

**PUT api/MedProductCard /WorkData**

|  |  |
| --- | --- |
| **Входящие параметры** | * WorkData\_id (N,O) – идентификтаор записи «Эксплуатационные данные» * MedProductCard\_id(N,Н) – идентификатор мед изделия * WorkData\_WorkPeriod(D,Н) – Период эксплуатации; Дата уникальна для МИ.Первое число месяца. * WorkData\_DayChange(N; Н) – Количество смен в сутки; * WorkData\_CountUse(N; Н) – Общее количество применений за период; * WorkData\_KolDay(N; Н) – Количество рабочих дней в периоде; |
| **Ответ** | Если изменяется одно из полей: WorkData\_CountUse, WorkData\_DayChange, WorkData\_KolDay, то производить перерасчет значения по формуле: WorkData\_AvgUse = WorkData\_CountUse/( WorkData\_DayChange\*WorkData\_KolDay) Округление до двух цифр после запятой. В ответе возвращать WorkData\_AvgUse. |

* + 1. Удаление свойств Медицинского изделия

**DELETE api/MedProductCardAttributes**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedProductCard\_id(N,O) – идентификатор мед изделия |
| **Ответ** | Необходимоудалить следующие записи:  Amortization where MedProductCard\_id = «входящий параметр»  MeasureFundCheck where MedProductCard\_id = «входящий параметр»  Downtime where MedProductCard\_id = «входящий параметр»  WorkData where MedProductCard\_id = «входящий параметр»  Consumables where MedProductCard\_id = «входящий параметр»  Если ничего не передано, то ошибка. |

* 1. Описание методов работы с данными организации
  2. 1. Поиск организации по параметрам

**GET api/OrgByPar**

|  |  |
| --- | --- |
| **Входящие параметры** | * Org\_id (N,У) – идентификтаор записи «Организация» * Org\_Code (N,У) –код организации; * Org\_Name(T,У) – наименование организации; * Org\_Nick (T,У) – краткое наименование организации;   Обязателен хотя бы один из параметров |
| **Ответ** | * Org\_id (N,O) – идентификтаор записи «Организация» * Org\_Code (N,O) –код организации; * Org\_Name(T,O) – наименование организации; * Org\_Nick (T,O) – краткое наименование организации; * OrgType\_id(N,O) – идентификатор типа организации; * UAddress\_id(N, У) - юридический адрес; |

* + 1. Создание записи «Организация»

**POST api/Org**

|  |  |
| --- | --- |
| **Входящие параметры** | * Org\_Code (N,O) –код организации. Формируется автоматически.; * Org\_Name(T,O) – наименование организации; * Org\_Nick (T,O) – краткое наименование организации. Устанавливается тоже значение, что и для Org\_Name; * OrgType\_id(N,O) – идентификатор типа организации. Устанавливаем значение 19 (Иные организации). * UAddress\_id(N, У) - юридический адрес. Обязателен, при создании записи организации-производителя; |
| **Ответ** | Org\_id (N,O) – идентификтаор записи «Организация» |

* + 1. Редактирование записи «Организация»

**PUT api/Org**

|  |  |
| --- | --- |
| **Входящие параметры** | * Org\_id (N,O) – идентификтаор записи «Организация» * Org\_Code (N,Н) – код организации; * Org\_Name(T,Н) – наименование организации; * Org\_Nick (T,Н) – краткое наименование организации; * OrgType\_id(N,Н) – идентификатор типа организации; * UAddress\_id(N, Н) - юридический адрес; |

* + 1. Получение записи «Адрес» для организации

**GET api/Org/Address**

|  |  |
| --- | --- |
| **Входящие параметры** | * Org\_id |
| **Ответ** | * Address\_Address(T,У) – строка адреса при ручном вводе. Заполняется если страна Россия. * KLCountry\_id (N,У) – идентификатор страны. Справочник стран. Заполняется, если страна не Россия * Address\_id (N,O) – идентификатор записи «Адрес» |

* + 1. Создание записи «Адрес»

**POST api/Org/Address**

|  |  |
| --- | --- |
| **Входящие параметры** | * Address\_Address(T,У) – строка адреса при ручном вводе. Заполняется если страна Россия. * KLCountry\_id (N,У) – идентификатор страны. Справочник стран. Заполняется, если страна не Россия |
| **Ответ** | Address\_id (N,O) – идентификатор записи «Адрес» |
| **Пример** |  |

* + 1. Редактирование записи «Адрес»

**PUT api/Org/Address**

|  |  |
| --- | --- |
| **Входящие параметры** | * Address\_id (N,O) – идентификатор записи «Адрес» * Address\_Address(T,У) – строка адреса при ручном вводе. Заполняется если страна Россия. * KLCountry\_id (N,У) – идентификатор страны. Справочник стран. Заполняется, если страна не Россия |

* + 1. Получение кода для новой организации

**GET api/Org/GenOrgCode**

|  |  |
| --- | --- |
| **Входящие параметры** |  |
| **Ответ** | * Org\_Code (N,O) - в ответе возвращать максимальный Org\_Code + 1 |

1. Электронная медицинская карта

Структура СЭМД:

* Заголовок;
* Эпикриз в стационаре выписной;
* Эпикриз в амбулатории;
* Направление.

Описание последовательности вызова методов приведено в п. 1.5.

1. 1. Описание методов работы с ТАП
2. 1. 1. Создание ТАП с первым посещением для поликлинического случая

**POST api/ EvnPLBase**

|  |  |
| --- | --- |
| **Входящие параметры** | *Данные талона*   * Person\_id (N,O) –идентификатор человека * EvnPL\_NumCard (T,О) № талона. Уникальный в рамках МО. * Lpu\_did (N,H)- направившая МО * Org\_did (N,H) - Направившая организация * EvnDirection\_id(N,Н) - идентификатор направления. * EvnDirection\_Num (N,H) –номер направления. * EvnDirection\_setDate (D,Н) – дата направления * LpuSection\_did (N,Н) –идентификатор направившего отделения. . * EvnPL\_IsFinish (N,О) Признак законченности случая. Возможные значения: 0 и 1, - где 0 – нет, 1 – да. * ResultClass\_id (N, У) – Результат обращения (Обязательное поле, если случай закончен); * ResultDeseaseType\_id (N,У) –Исход обращения (Обязательное поле, если случай закончен); * Diag\_lid (N,У) – заключительный диагноз Обязательное поле, если случай закончен);   *Данные посещения*   * Evn\_setDT (D, O) – Дата и время посещения; * VizitClass\_id (N,H) – Идентификатор вида посещения; * LpuSection\_id (N, O) идентификатор отделения ЛПУ * MedStaffFact\_id (N, O) Специальность врача: идентификатор справочника медицинских работников * MedStaffFact\_sid (N,О) средний мед персонал (в ТЗ «Посещение» поле обязательное) * TreatmentClass\_id (N,O) Вид обращения * ServiceType\_id (N,O)Место обслуживания * VizitType\_id (N,O) цель посещения * PayType\_id (N,O) Тип оплаты * Mes\_id (N,У) МЭС (передается только для первого посещения) * UslugaComplex\_uid (N,У) –код посещения; Обязательно если вид оплаты ОМС * EvnVizitPL\_Time(N,H) Время (мин); * ProfGoal\_id (N,H) Цель профосмотра * DispClass\_id (N,H) В рамках дисп./мед.осмотра * EvnPLDisp\_id (N,H) Карта дисп./мед осмотра * PersonDisp\_id (N,H) Идентификатор карты дисп учета * Diag\_id (N,О) Основной диагноз (В ТЗ поле не обязательное, на форме обязательное) * DeseaseType\_id (N,O) характер заболевания (если заполнен диагноз, то поле обязательное) * Diag\_agid (N,H) осложнение * RankinScale\_id (N,H) Значение по шкале Рэнкина * HomeVisit\_id (N,У) –Идентификатор посещения на дому * MedicalCareKind\_id(N,О) - Идентификатор вида медицинской помощи |
| **Ответ** | * EvnVizitPL\_id (N, О) - Идентификатор посещения * EvnPLBase\_id (N,О) – Идентификатор ТАП |

* + 1. Добавление посещения в ТАП Поликлинический случай

**POST api/ EvnVizitPL**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnPLBase\_id (N,О) – Идентификатор ТАП   *Данные посещения*   * Evn\_setDT (D, O) – Дата и время посещения; * VizitClass\_id (N,H) – Идентификатор вида посещения; * LpuSection\_id (N, O) идентификатор отделения ЛПУ * MedStaffFact\_id (N, O) Специальность врача: идентификатор справочника медицинских работников * MedStaffFact\_sid (N,О) средний мед персонал (в ТЗ «Посещение» поле обязательное) * TreatmentClass\_id (N,O) Вид обращения * ServiceType\_id (N,O)Место обслуживания * VizitType\_id (N,O) цель посещения * PayType\_id (N,O) Тип оплаты * Mes\_id (N,У) МЭС (МЭС для второго и следующих посещений в рамках одного ТАП должен совпадать с МЭС первого посещения) **Регион: Свердловская область** * UslugaComplex\_uid (N,У) –код посещения; Обязательно если вид оплаты ОМС * EvnVizitPL\_Time(N,H) Время (мин); * ProfGoal\_id (N,H) Цель профосмотра * DispClass\_id (N,H) В рамках дисп./мед.осмотра * EvnPLDisp\_id (N,H) Карта дисп./мед осмотра * PersonDisp\_id (N,H) Идентификатор карты дисп учета * Diag\_id (N,О) Основной диагноз (В ТЗ поле не обязательное, на форме обязательное) * DeseaseType\_id (N,O) характер заболевания (если заполнен диагноз, то поле обязательное) * Diag\_agid (N,H) осложнение * RankinScale\_id (N,H) Значение по шкале Рэнкина * HomeVisit\_id (N,У) –Идентификатор посещения на дому * MedicalCareKind\_id(N,О) - Идентификатор вида медицинской помощи |
| **Ответ** | **Успешный ответ:**   * EvnVizitPL\_id (N, О) - Идентификатор посещения; * Evn\_id (N, Н) – Идентификатор случая   **Ошибки:**   * Если МЭС, переданный для второго и последующих посещений, не совпадает с МЭС первого посещения в рамках одного ТАП, то возвращается ошибка: «МЭС посещения не сопадает с МЭС первого посещения в рамках одного ТАП». (**Регион: Свердловская область)** |
| **Пример** | Запрос ответ |

* + 1. Редктирование ТАП Поликлинический случай

**PUT api/ EvnPLBase**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnPLBase\_id (N,О) – Идентификатор ТАП * Lpu\_did (N,H)- направившая МО * Org\_did (N,H) - Направившая организация * EvnDirection\_id(N,Н) - идентификатор направления. * EvnDirection\_Num (N,H) –номер направления. * EvnDirection\_setDate (D,Н) – дата направления * LpuSection\_did (N,Н) –идентификатор направившего отделения. . * EvnPL\_IsFinish (N,Н) Признак законченности случая. (Если ТАП новый, то поле обязательное) Возможные значения: 0 и 1, - где 0 – нет, 1 – да. * ResultClass\_id (N, У) – Результат обращения (Обязательное поле, если случай закончен); * ResultDeseaseType\_id (N,У) – Исход обращения (Обязательное поле, если случай закончен); * Diag\_lid (N,У) – заключительный диагноз Обязательное поле, если случай закончен); |
| **Ответ** | **Успешный ответ**  Возвращается пустой ответ с кодом ошибки 0 |

* + 1. Редактирование Посещения поликлинический случай

**PUT api/EvnVizitPL**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnVizitPL\_id (N, О) - Идентификатор посещения * Evn\_id (N, Н) – Идентификатор случая посещения; * Evn\_setDT (D, O) – Дата и время посещения; * VizitClass\_id (N,H) – Идентификатор вида посещения; * LpuSection\_id (N, O) идентификатор отделения ЛПУ; * MedStaffFact\_id (N, Н) Специальность врача: идентификатор справочника медицинских работников; * MedStaffFact\_sid (N,O) средний мед персонал (в ТЗ «Посещение» поле обязательное); * TreatmentClass\_id (N,O) Вид обращения; * ServiceType\_id (N, O)Место обслуживания; * VizitType\_id (N, O) цель посещения; * PayType\_id (N, O) Тип оплаты; * Mes\_id (N, Н) МЭС (МЭС для второго и следующих посещений в рамках одного ТАП должен совпадать с МЭС первого посещения) **Регион: Свердловская область** * UslugaComplex\_uid (N,Н) –код посещения; Обязательно если вид оплаты ОМС; * EvnVizitPL\_Time(N, H) Время (мин); * ProfGoal\_id (N, H) Цель профосмотра; * DispClass\_id (N,H) В рамках дисп./мед.осмотра; * EvnPLDisp\_id (N,H) Карта дисп./мед осмотра; * PersonDisp\_id (N,H) Идентификатор карты дисп учета * Diag\_id (N,Н) Основной диагноз (В ТЗ поле не обязательное, на форме обязательное); * DeseaseType\_id (N,Н) характер заболевания (если заполнен диагноз, то поле обязательное); * Diag\_agid (N,H) осложнение; * RankinScale\_id (N,H) Значение по шкале Рэнкина; * HomeVisit\_id (N,У) – Идентификатор посещения на дому; * MedicalCareKind\_id(N, O) - Идентификатор вида медицинской помощи. |
| **Ответ** | **Успешный ответ**   * Возвращается пустой ответ с кодом ошибки 0   **Ошибки:**   * Если МЭС, переданный для второго и последующих посещений, не совпадает с МЭС первого посещения в рамках одного ТАП, то возвращается ошибка: «МЭС посещения не сопадает с МЭС первого посещения в рамках одного ТАП». (**Регион: Свердловская область)** |

* + 1. Получить ТАП

**GET api/ EvnPLBase**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnPLBase\_id (N,Н) – Идентификатор ТАП; * Person\_id (N,Н) – идентификатор человека; * EvnPL\_NumCard (T,Н) - № талона. |
| **Ответ** | Если не передан ни один параметр (не считая Lpu\_id), то ошибка.  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список ТАП с информацией:   * EvnPLBase\_id (N,Н) – Идентификатор ТАП; * Person\_id (N,O) идентификатор человека; * EvnPL\_NumCard (T,О) № талона. * Lpu\_did (N,H)- направившая МО * Org\_did (N,H) - Направившая организация * EvnDirection\_id(N,Н) - идентификатор направления. * EvnDirection\_Num (N,H) –номер направления. * EvnDirection\_setDate (D,Н) – дата направления * LpuSection\_did (N,Н) –идентификатор направившего отделения. . * EvnPL\_IsFinish (N,О) Признак законченности случая. (Если ТАП новый, то поле обязательное) * ResultClass\_id (N, У) – Результат обращения (Обязательное поле, если случай закончен); * ResultDeseaseType\_id (N,У) –Исход обращения (Обязательное поле, если случай закончен); * Diag\_lid (N,У) – заключительный диагноз Обязательное поле, если случай закончен); |

* + 1. Получить посещение

**GET api/ EvnVizitPL**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnVizitPL\_id (N, Н) - Идентификатор посещения; * Evn\_id (N, Н) – Идентификатор случая; * EvnPLBase\_id (N,Н) – Идентификатор ТАП; * Evn\_setDT (D, Н) – Дата и время посещения. |
| **Ответ** | Если не передан ни один параметр, то ошибка.  Если не найдено ни одной записи, то ответ «0».  Если в параметре Evn\_setDT указано значение времени «00:00:00.000», то поиск посещения производится на протяжении всей указанной даты.  Если хотя бы одна запись найдена, то успешный ответ – список посещений с информацией:   * EvnVizitPL\_id (N, Н) - Идентификатор посещения; * Evn\_id (N, О) – Идентификатор случая; * EvnPLBase\_id (N,У) – Идентификатор ТАП * Evn\_setDT (D, O) – Дата и время посещения; * VizitClass\_id (N,H) – Идентификатор вида посещения; * LpuSection\_id (N, O) идентификатор отделения ЛПУ * MedStaffFact\_id (N, O) Специальность врача: идентификатор справочника медицинских работников * MedStaffFact\_sid (N,О) средний мед персонал (в ТЗ «Посещение» поле обязательное) * TreatmentClass\_id (N,O) Вид обращения * ServiceType\_id (N,O)Место обслуживания * VizitType\_id (N,O) цель посещения * PayType\_id (N,O) Тип оплаты * Mes\_id (N,У) МЭС (передается только для первого посещения) * UslugaComplex\_uid (N,У) –код посещения; Обязательно если вид оплаты ОМС * EvnVizitPL\_Time(N,H) Время (мин); * ProfGoal\_id (N,H) Цель профосмотра * DispClass\_id (N,H) В рамках дисп./мед.осмотра * EvnPLDisp\_id (N,H) Карта дисп./мед осмотра * PersonDisp\_id (N,H) Идентификатор карты дисп учета. * Diag\_id (N,О) Основной диагноз (В ТЗ поле не обязательное, на форме обязательное) * DeseaseType\_id (N,O) характер заболевания (если заполнен диагноз, то поле обязательное) * Diag\_agid (N,H) осложнение * RankinScale\_id (N,H) Значение по шкале Рэнкина * HomeVisit\_id (N,У) –Идентификатор посещения на дому * MedicalCareKind\_id(N,О) - Идентификатор вида медицинской помощи |

* + 1. Получить список посещений для ТАП

**GET api/EvnVizitPL\_List**

|  |  |
| --- | --- |
| **Входящие параметры** | EvnPLBase\_id (N,У) – Идентификатор ТАП |
| **Ответ** | Если не передан ни параметр, то ошибка.  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список посещений со следующими полями:   * EvnVizitPL\_id (N, О) - Идентификатор посещения |

* + 1. Получить данные ТАП (стоматология) по идентификатору

**GET api/EvnPLStom**

|  |  |
| --- | --- |
| **Входящие параметры** | EvnPLStom\_id(N,O) –идентификатор ТАП (стоматология) |
| **Ответ** | Person\_id (N,O) –идентификатор человека (пациента);  Lpu\_id (N,O) –идентификатор МО;  Date(D,O) – дата начала случая;  NumCard (T,O) – номер талона;  IsFinish (N,O) –признак законченности случая;  ResultClass\_id(N,У) – Идентификатор результата лечения;  ResultDeseaseType\_id(N,У) – Идентификатор исхода;  EvnPLStom\_UKL (N,У) - УКЛ  Diag\_lid(N,У) – Идентификатор заключтительного диагноза; |

* + 1. Получить список посещений для ТАП (стоматология)

**GET api/EvnVizitPLStom\_List**

|  |  |
| --- | --- |
| **Входящие параметры** | EvnPLStom\_id(N,O) –идентификатор ТАП (стоматология) |
| **Ответ** | Список посещений, содержпщий следующие элементы:  EvnVizitPLStom\_id (N,O) – идентификатор посещения; |

* + 1. Получить список заболеваний для ТАП (стоматология)

**GET api/EvnDiagPLStom\_List**

|  |  |
| --- | --- |
| **Входящие параметры** | EvnPLStom\_id(N,O) –идентификатор ТАП (стоматология) |
| **Ответ** | Список, содержащий следующие элементы:  EvnDiagPLStom\_id (N,O) –Идентификатор заболевания. |
| **Пример** | Запрос ответ |

* + 1. Получить посещение по идентификатору(стоматология)

**GET api/EvnVizitPLStom**

|  |  |
| --- | --- |
| **Входящие параметры** | EvnVizitPLStom\_id (N,O) – идентификатор посещения; |
| **Ответ** | EvnVizitPLStom\_setDate (D,O) – дата посещения;  LpuSection\_id (N,O) – идентификатор отделения;  MedStaffFact\_id(N,O) – идентификатор врача;  TreatmentClass\_id (N,O) –вид обращения;  ServiceType\_id(N,O) – место;  VizitType\_id (N,O) –цель посещения;  PayType\_id(N,O) – вид оплаты;  MesEkb\_id(N,O) – идентификатор МЭС;  UslugaComplex\_uid(N,O) –код посещения;  MedicalCareKind\_id (N,О) – Идентификатор вида медицинской помощи. |

* + 1. Получить заболевание по идентификатору (стоматология)

**GET api/EvnDiagPLStom**

|  |  |
| --- | --- |
| **Входящие параметры** | EvnDiagPLStom\_id (N,O) –Идентификатор заболевания |
| **Ответ** | Diag\_id(N,O) –идентификатор основного диагноза;  DeseaseType\_id(N,O) –Характер диагноза;  EvnDiagPLStom\_IsClosed (N,O) – признак «Заболевание закрыто» |

* + 1. Создать ТАП стоматологический

**POST api/EvnPLStom**

|  |  |
| --- | --- |
| **Входящие параметры** | *Данные ТАП*  Person\_id (N,O) –идентификатор человека (пациента);  Lpu\_id (N,O) –идентификатор МО;  Date(D,O) – дата начала случая;  NumCard (T,O) – номер талона;  IsFinish (N,O) –признак законченности случая; Возможные значения: 0 и 1, - где 0 – нет, 1 – да  ResultClass\_id(N,У) – Идентификатор результата лечения;  ResultDeseaseType\_id(N,У) – Идентификатор исхода;  EvnPLStom\_UKL (N,У) - УКЛ  Diag\_lid(N,У) – Идентификатор заключтительного диагноза;  *Данные посещения*  EvnVizitPLStom\_setDate (D,O) – дата посещения;  LpuSection\_id (N,O) – идентификатор отделения;  MedStaffFact\_id(N,O) – идентификатор врача;  TreatmentClass\_id (N,O) –вид обращения;  ServiceType\_id(N,O) – место;  VizitType\_id (N,O) –цель посещения;  PayType\_id(N,O) – вид оплаты;  MesEkb\_id(N,O) – идентификатор МЭС;  UslugaComplex\_uid(N,O) –код посещения;  *Данные заболевания (заболевание создается в рамках первого посещения)*  Diag\_id(N,У) –идентификатор основного диагноза;  DeseaseType\_id(N,У) –Характер диагноза;  Tooth\_Code(N,H) –зуб;  EvnDiagPLStom\_IsClosed (N, О) – признак «Заболевание закрыто». Возможные значения: 0 и 1, - где 0 – нет, 1 – да  MedicalCareKind\_id(N,О) - Идентификатор вида медицинской помощи |
| **Ответ** | EvnPLStom\_id(N,O) –идентификатор ТАП (стоматология)  EvnVizitPLStom\_id (N,O) – идентификатор посещения;  EvnDiagPLStom\_id (N,У) –Идентификатор заболевания. |

* + 1. Установка заболевания (ТАП стоматологический)

**POST api/EvnDiagPLStom**

|  |  |
| --- | --- |
| **Входящие параметры** | EvnVizitPLStom\_id (N,O) – идентификатор посещения, в котором заболевание создано  Diag\_id(N,O) –идентификатор основного диагноза;  DeseaseType\_id(N,O) –Характер диагноза;  EvnDiagPLStom\_IsClosed (N,O) – признак «Заболевание закрыто». Возможные значения: 0 и 1, - где 0 – нет, 1 – да |
| **Ответ** | EvnDiagPLStom\_id (N,O) –Идентификатор заболевания. |

* + 1. Создать посещение (ТАП стоматологический)

**POST api/EvnVizitPLStom**

|  |  |
| --- | --- |
| **Входящие параметры** | EvnPLStom\_id(N,O) –идентификатор ТАП (стоматология)  EvnVizitPLStom\_setDate (D,O) – дата посещения;  LpuSection\_id (N,O) – идентификатор отделения;  MedStaffFact\_id(N,O) – идентификатор врача;  TreatmentClass\_id (N,O) –вид обращения;  ServiceType\_id(N,O) – место;  VizitType\_id (N,O) –цель посещения;  PayType\_id(N,O) – вид оплаты;  MesEkb\_id(N,O) – идентификатор МЭС;  UslugaComplex\_uid(N,O) –код посещения;  MedicalCareKind\_id (N,О) – Идентификатор вида медицинской помощи. |
| **Ответ** | EvnVizitPLStom\_id (N,O) – идентификатор посещения; |

* + 1. Изменить посещение (ТАП стоматологический)

**PUT api/EvnVizitPLStom**

|  |  |
| --- | --- |
| **Входящие параметры** | EvnVizitPLStom\_id (N,O) – идентификатор посещения;  EvnVizitPLStom\_setDate (D,H) – дата посещения;  LpuSection\_id (N,H) – идентификатор отделения;  MedStaffFact\_id(N,H) – идентификатор врача;  TreatmentClass\_id (N,H) –вид обращения;  ServiceType\_id(N,H) – место;  VizitType\_id (N,H) –цель посещения;  PayType\_id(N,H) – вид оплаты;  MesEkb\_id(N,H) – идентификатор МЭС;  UslugaComplex\_uid(N,H) –код посещения;  MedicalCareKind\_id (N,H) – Идентификатор вида медицинской помощи. |
| **Ответ** | **Успешный ответ**  Возвращается пустой ответ с кодом ошибки 0 |

* + 1. Изменить заболевание (ТАП стоматологический)

**PUT api/EvnDiagPLStom**

|  |  |
| --- | --- |
| **Входящие параметры** | EvnVizitPLStom\_id (N,O) – идентификатор посещения, в котором заболевание изменяется  EvnDiagPLStom\_id (N,O) –Идентификатор заболевания.  Diag\_id(N,Н) –идентификатор основного диагноза;  DeseaseType\_id(N,Н) –Характер диагноза;  EvnDiagPLStom\_IsClosed (N,Н) – признак «Заболевание закрыто» |
| **Ответ** | **Успешный ответ**  Возвращается пустой ответ с кодом ошибки 0 |

* + 1. Получить информацию об оказанной услуги по идентификатору (стоматология)

**GET api/EvnUslugaStom**

|  |  |
| --- | --- |
| **Входящие параметры** | EvnUslugaStom\_id(N,O) – идентификатор услуги |
| **Ответ** | EvnVizitPLStom\_id (N,O) – идентификатор посещения;  EvnDiagPLStom\_id (N,O) –Идентификатор заболевания.  MedStaffFact\_id (N,O) – идентификатор врача, выполневшего услугу;  PayType\_id(N,O) – вид оплаты;  UslugaCategory\_id(N,O) – идентификатор категории услуги;  UslugaComplex\_id(N,O) –идентификатор услуги; |

* + 1. Получить список оказанных услуг в рамках заболевания (стоматология)

**GET api/EvnUslugaStom\_List**

|  |  |
| --- | --- |
| **Входящие параметры** | EvnDiagPLStom\_id (N,O) –Идентификатор заболевания. |
| **Ответ** | Список записей, содержащих следующие элементы  EvnUslugaStom\_id(N,O) – идентификатор услуги |

* + 1. Получить список оказанных услуг в рамках посещения (стоматология)

**GET api/EvnUslugaStom\_List**

|  |  |
| --- | --- |
| **Входящие параметры** | EvnVizitPLStom\_id (N,O) – идентификатор посещения; |
| **Ответ** | Список записей, содержащих следующие элементы  EvnUslugaStom\_id(N,O) – идентификатор услуги |

* + 1. Создать услугу в Стоматологии

**POST api/EvnUslugaStom**

|  |  |
| --- | --- |
| **Входящие параметры** | EvnVizitPLStom\_id (N,O) – идентификатор посещения;  EvnDiagPLStom\_id (N,O) –Идентификатор заболевания.  MedStaffFact\_id (N,O) – идентификатор врача, выполневшего услугу;  PayType\_id(N,O) – вид оплаты;  UslugaComplex\_id(N,O) –идентификатор услуги; |
| **Ответ** | EvnUslugaStom\_id(N,O) – идентификатор услуги |

* + 1. Изменить услугу в Стоматологии

**PUT api/EvnUslugaStom**

|  |  |
| --- | --- |
| **Входящие параметры** | EvnUslugaStom\_id(N,O) – идентификатор услуги  EvnVizitPLStom\_id (N,Н) – идентификатор посещения;  EvnDiagPLStom\_id (N,Н) –Идентификатор заболевания.  MedStaffFact\_id (N,Н) – идентификатор врача, выполневшего услугу;  PayType\_id(N,Н) – вид оплаты;  UslugaCategory\_id(N,Н) – идентификатор категории услуги;  UslugaComplex\_id(N,Н) –идентификатор услуги; |
| **Ответ** | **Успешный ответ**  Возвращается пустой ответ с кодом ошибки 0 |

* + 1. Добавить сопутствующий диагноз для заболевания

**POST api/DiagPLStomSop**

|  |  |
| --- | --- |
| **Входящие параметры** | EvnVizitPLStom\_id (N,O) – идентификатор посещения;  EvnDiagPLStom\_id (N,O) –Идентификатор заболевания;  Diag\_id(N,O) – Идентификатор диагноза;  DeseaseType\_id(N,O) – Идентификатор Характера заболевания |
| **Ответ** | DiagPLStomSop\_id(N,O) – идентификатор сопутствующего диагноза |

* + 1. Получить список сопутствующих диагнозов для заболевания

**GET api/DiagPLStomSop\_List**

|  |  |
| --- | --- |
| **Входящие параметры** | EvnDiagPLStom\_id (N,O) –Идентификатор заболевания; |
| **Ответ** | Список, состоящий из следующих элементов:  DiagPLStomSop\_id(N,O) – идентификатор сопутствующего диагноза |

* + 1. Получить сопутствующий диагноз для заболевания по идентификатору

**GET api/DiagPLStomSop**

|  |  |
| --- | --- |
| **Входящие параметры** | DiagPLStomSop\_id(N,O) – идентификатор сопутствующего диагноза |
| **Ответ** | Diag\_id(N,O) – Идентификатор диагноза;  DeseaseType\_id(N,O) – Идентификатор Характера заболевания |

* + 1. Изменить сопутствующий диагноз для заболевания

**PUT api/DiagPLStomSop**

|  |  |
| --- | --- |
| **Входящие параметры** | DiagPLStomSop\_id(N,O) – идентификатор сопутствующего диагноза  Diag\_id(N, H) – Идентификатор диагноза;  DeseaseType\_id(N, H) – Идентификатор Характера заболевания |
| **Ответ** | **Успешный ответ**  Возвращается пустой ответ с кодом ошибки 0 |

* + 1. Получение информации по ТАП (закрытые/открытые)

**GET api/ EvnPLList**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnPL\_IsFinish (N,O) – признак ТАП «Случай закончен». Принимает значения 0 – нет, 1 – да. * Lpu\_id (N,Н) – идентификатор МО; |
| **Ответ** | Возвращает массив записей с набором атрибутов:   * EvnPL\_id (N,O) –идентификатор ТАП * Person\_id (N,O) – идентификатор человека; * Lpu\_id (N,О) – идентификатор МО; * EvnDirection\_id(N,Н) - идентификатор направления. * Lpu\_did (N,H)- направившая МО * Org\_did (N,H) - Направившая организация * Diag\_did (N,Н) –диагноз направившего учреждения * EvnDirection\_Num (N,H) –номер направления. * EvnDirection\_setDate (D,Н) – дата направления * Diag\_lid (N,O) – заключительный диагноз Обязательное поле |
| **Пример** |  |

* + 1. Получение информации по ТАП (стоматология) (закрытые/открытые)

**GET api/EvnVizitPLStomList**

|  |  |
| --- | --- |
| **Входящие параметры** | * IsFinish (N,O) –признак законченности случая. Возможные значения: 0 и 1, - где 0 – нет, 1 – да. * Lpu\_id (N,Н) – идентификатор МО; |
| **Ответ** | Возвращает массив записей с набором атрибутов:   * EvnPLStom\_id (N,O) –идентификатор ТАП Стоматология * Person\_id (N,O) – идентификатор человека; * Lpu\_id (N,О) – идентификатор МО; * Date(D,O) – дата начала случая; * EvnDirection\_id(N,Н) - идентификатор направления. * Lpu\_did (N,H)- направившая МО * Org\_did (N,H) - Направившая организация * Diag\_did (N,Н) –диагноз направившего учреждения * EvnDirection\_Num (N,H) –номер направления. * EvnDirection\_setDate (D,Н) – дата направления * Diag\_lid(N,У) – Идентификатор заключительного диагноза; |
| **Пример** |  |

* + 1. Получение информации по случаю амбулаторно-поликлинического лечения

**GET api/EvnPLBaseInfo**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnPLBase\_id (N,O) – Идентификатор ТАП; * EvnVizitPL\_id (N, O) - Идентификатор посещения; |
| **Ответ** | Успешный ответ:   * Diag\_fid (N, H) – Идентификатор предварительного диагноза; * Diag\_Code (N, H) – Код МКБ-10 предварительного диагноза; * Diag\_Name (T, H) – Наименование предварительного диагноза; * PreHospTrauma\_id (N, H) – Идентификатор вида травмы (внешнего воздействия) * TraumaType\_Code (N, H) – Код типа травмы; * TraumaType\_Name (T, H) – Наименование типа травмы; * TraumaClass\_Code (N, H) – Код класса травмы; * TraumaClass\_Name (T, H) – Наименование класса травмы; * EvnPL\_IsUnport (N, H) – Признак нетранспортабельности   Список назначений с типом назначения «Режим»: массив EvnPrescrRegimeList{   * EvnPrescrRegime\_id (N, H) – Идентификатор c типом назначения «Режим» * PrescriptionRegimeType\_Code (N, H) – Код типа режима; * PrescriptionRegimeType\_Name (T, H) – Наименование типа режима; * EvnPrescr\_setDate (D, H) – Дата начала режима; * EvnPrescr\_dayNum (N, H) – Продолжительность режима (дни); * EvnPrescrDescr (T, H) – Комментарий к назначению режима;   }  Список назначений с типом назначения «Диета»: массив EvnPrescrDietList{   * EvnPrescrDiet\_id (N,H) – Идентификатор c типом назначения «Диета»; * PrescriptionDietType\_Code (N, H) – Код типа диеты; * PrescriptionDietType\_Name (T, H) – Наименование типа диеты; * EvnPrescr\_setDate (D, H) – Дата начала диеты; * EvnPrescr\_dayNum (N, H) – Продолжительность диеты (дни); * EvnPrescrDescr (T, H) – Комментарий к назначению диеты;   }  Список назначений с типом назначения «Лекарственное лечение» (метод получения назначений лекарственных средств: GET api/EvnPrescrTreat): массив EvnPrescrTreatList{   * EvnPrescrTreat\_id (N,H) – Идентификатор c типом назначения «Лекарственное лечение»   }  Список назначений с типом назначения «Манипуляции и процедуры»: массив EvnPrescrProcList{   * EvnPrescrProc\_id (N,H) – Идентификатор c типом назначения «Манипуляции и процедуры»; * TimeTableGraf\_id (N, H) – Идентификатор поликлинической бирки; * TimeTableStac\_id (N, H) – Идентификатор стационарной бирки; * EvnQueue\_id (N, H) – Идентификатор постановки в очередь; * UslugaComplex\_id (N, H) – Идентификатор услуги (значение справочника dbo.UslugaComplex); * UslugaComplex\_Name (Т, H) – Наименование услуги; * EvnPrescr\_setDate (D, H) – Дата начала процедур; * EvnCourse\_id (N, H) – Идентификатор курса; * EvnCourse\_MaxCountDay (N, H) – Количество повторов в сутки; * EvnCourse\_Duration (N, H) – Продолжительность процедур; * DurationType\_id (N, H) – Тип продолжительности процедуры (значение справочника dbo.DurationType); * EvnCourse\_ContReception (N, H) – Непрерывная продолжительность процедур; * DurationType\_recid (N, H) – Тип непрерывной продолжительности (значение справочника dbo.DurationType); * EvnCourse\_Interval (N, Н) – Продолжительность перерыва; * DurationType\_intid (N, H) – Тип продолжительности перерыва (значение справочника dbo.DurationType); * EvnPrescr\_IsCito (N, H) – Признак "Cito" (0 – Нет, 1 - Да); * EvnDirection\_id (N, H) – Идентификатор направления по назначению; * EvnPrescr\_IsExec (N, H) – Признак "Выполнено" (0 – Нет, 1 - Да);   }  Список назначений с типом назначения «Лабораторная диагностика»: массив EvnPrescrLabList{   * EvnPrescr\_id (N, H) – Идентификатор назначения; * PrescriptionType\_id (N, H) – Идентификатор типа назначения; * PrescriptionType\_Name (T, H) – Наименование типа назначения; * TimeTableGraf\_id (N, H) – Идентификатор поликлинической бирки; * TimeTableStac\_id (N, H) – Идентификатор стационарной бирки; * TimeTableMedService\_id (N, H) – Идентификатор лабораторной бирки; * TimeTableResource\_id (N, H) – Идентификатор диагностической бирки; * EvnQueue\_id (N, H) – Идентификатор постановки в очередь; * EvnPrescr\_IsCito (N, H) – Признак "Cito" (0 – Нет, 1 - Да);   Список услуг: UslugaComplexList{   * UslugaComplex\_id (N, H) – Идентификатор услуги (значение справочника dbo.UslugaComplex); * UslugaComplex\_Name (Т, H) – Наименование услуги; * DefectState\_Name (T, H) – Наименование по наличию брака пробы в исследовании (значение справочника lis.DefectState); * DefectCauseType\_Name (T, H) – Наименование причины брака пробы в исследовании (значение справочника lis.DefectCauseType);   }   * EvnPrescr\_IsExec (N, H) – Признак "Выполнено" (0 – Нет, 1 - Да); * EvnDirection\_id (N, H) – Идентификатор направления по назначению.   }  Список назначений с типами назначения «Оперативное лечение», «Функциональная диагностика», Консультационная услуга»: массив EvnPrescrOperDiagConsList{   * EvnPrescr\_id (N, H) – Идентификатор назначения; * PrescriptionType\_id (N, H) – Идентификатор типа назначения; * PrescriptionType\_Name (T, H) – Наименование типа назначения; * TimeTableGraf\_id (N, H) – Идентификатор поликлинической бирки; * TimeTableStac\_id (N, H) – Идентификатор стационарной бирки; * TimeTableMedService\_id (N, H) – Идентификатор лабораторной бирки; * TimeTableResource\_id (N, H) – Идентификатор диагностической бирки; * EvnQueue\_id (N, H) – Идентификатор постановки в очередь; * EvnPrescr\_IsCito (N, H) – Признак "Cito" (0 – Нет, 1 - Да); * UslugaComplex\_id (N, H) – Идентификатор услуги (значение справочника dbo.UslugaComplex); * UslugaComplex\_Name (Т, H) – Наименование услуги; * EvnPrescr\_IsExec (N, H) – Признак "Выполнено" (0 – Нет, 1 - Да); * EvnDirection\_id (N, H) – Идентификатор направления по назначению.   }   * XmlZhaloby\_Data (T, H) – Текст поля «Жалобы» базового шаблона осмотра; * XmlAnamnezZhizni\_Data (T, H) – Текст поля «Анамнез жизни» базового шаблона осмотра; * XmlAnamnezZabolev\_Data (T, H) – Текст поля «Анамнез заболевания» базового шаблона осмотра; * XmlObjStatus\_Data (T, H) – Текст поля «Объективный статус» базового шаблона осмотра; * XmlLocalStatus\_Data (T, H) – Текст поля «Локальный статус» базового шаблона осмотра; * XmlDiag\_Data (T, H) – Текст поля «Диагноз основной (расшифровка)» базового шаблона осмотра; * XmlRecomend\_Data (T, H) – Текст поля «Рекомендации, назначения» базового шаблона осмотра; * XmlZakluchenie\_Data (T, H) – Текст поля «Заключение» базового шаблона осмотра; * XmlSoputZabol\_Data (T, H) – Текст поля «Перенесенные сопутствующие заболевания» базового шаблона осмотра; * XmlRentgen\_Data (T, H) – Текст поля «Рентгеновские, лабораторные исследования» базового шаблона осмотра; * XmlNastavlenia\_Data (T, H) – Текст поля «Наставления» базового шаблона осмотра. |
| **Пример** | Запрос ответ |

* 1. Описание методов работы с услугами
  2. 1. Получение списка выполненных услуг

**GET api/EvnUslugaList**

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_pid (N, О) – Идентификатор случая-родителя (движение в КВС, посещения в ТАП) |
| **Ответ** | Успешный ответ – список услуг:   * EvnUsluga\_id (N, О) – Идентификатор оказания услуги; * Evn\_id (N, О) – Идентификатор случая оказания услуги (значение сущности dbo.Evn); * Evn\_pid (N, О) – Идентификатор случая-родителя (движение в КВС, посещения в ТАП) (значение сущности dbo.Evn); * EvnClass\_id (N, О) – Класс события (значение справочника dbo.EvnClass); * Evn\_setDT (DT, О) – Дата и время начала выполнения услуги; * Evn\_disDT (DT, О) – Дата и время окончания выполнения услуги; * UslugaComplex\_id (N, О) – Услуга (значение справочника dbo.UslugaComplex) |

* + 1. Получение информации по оперативной услуге

**GET api/EvnUslugaOper**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnUsluga\_id (N, Н) – Идентификатор оказания услуги при условии, что Класс события = «43. Оперативная услуга»; * Evn\_pid (N, Н) – Идентификатор случая-родителя (движение в КВС, посещения в ТАП) (значение сущности dbo.Evn); * Evn\_setDT (DT, Н) – Дата и время начала выполнения услуги; * UslugaComplex\_id (N, Н) – Услуга (значение справочника dbo.UslugaComplex). |
| **Ответ** | Если не передан ни один параметр, то ошибка.  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список оперативных услуг с информацией:   * EvnUsluga\_id (N, О) – Идентификатор оказания услуги; * EvnUslugaOper\_id (N, О) – Идентификатор оказания оперативной услуги; * Evn\_id (N, О) – Идентификатор случая оказания услуги (значение сущности dbo.Evn); * Evn\_pid (N, О) – Идентификатор случая-родителя (движение в КВС, посещения в ТАП) (значение сущности dbo.Evn); * EvnClass\_id (N, О) – Класс события (значение справочника dbo.EvnClass); * Evn\_setDT (DT, О) – Дата и время начала выполнения услуги; * Evn\_disDT (DT, О) – Дата и время окончания выполнения услуги; * UslugaPlace\_id (N, О) – Место выполнения (значение справочника dbo.UslugaPlace); * LpuSection\_id (N, У/О) – Отделение МО (значение сущности dbo.LpuSection); * Lpu\_id (N, У/О) – МО (значение справочника dbo.Lpu); * Org\_id (N, У/О) – Другая организация (значение справочника dbo.Org); * LpuSectionProfile\_id(N, У/О) – Профиль отделения МО (значение справочника dbo.LpuSectionProfile); * MedSpecOms\_id (N, У/О) – Специальность (значение справочника dbo.MedSpecOms); * MedStaffFact\_id (N, У/О) – Место работы врача, оказавшего услугу (значение сущности dbo.MedStaffFact); * PayType\_id (N, О) – Вид оплаты (значение справочника dbo.PayType); * EvnPrescr\_id (N, Н) – Назначение (значение сущности dbo.EvnPrescr); * DiagSetClass\_id (N, О) – Тип диагноза (значение справочника dbo.DiagSetClass); * Diag\_id (N, О) – Диагноз (значение справочника dbo.Diag); * UslugaCategory\_id (N, О) – Категория услуги (значение справочника dbo.UslugaCategory); * UslugaComplex\_id (N, О) – Услуга (значение справочника dbo.UslugaComplex); * EvnUsluga\_Price (N, Н) – Цена услуги; * OperType\_id (N, О) – Тип операции (значение справочника dbo.OperType); * OperDiff\_id (N, О) – Категория сложности (значение справочника dbo.OperDiff); * TreatmentConditionsType\_id (N, Н) – Условие лечения (значение справочника dbo.TreatmentConditionsType); * EvnUslugaOper\_IsVMT (N, Н) – Признак "Применение ВМТ"; * EvnUslugaOper\_IsMicrSurg (N, Н) – Признак "Микрохирургическая"; * EvnUslugaOper\_IsOpenHeart (N, Н) – Признак "На открытом сердце"; * EvnUslugaOper\_IsArtCirc (N, Н) – Признак "С искусственным кровообращением"; * EvnUslugaOper\_IsEndoskop (N, О) – Признак "Использование эндоскопической аппаратуры"; * EvnUslugaOper\_IsLazer (N, О) – Признак "Использование лазерной аппаратуры"; * EvnUslugaOper\_IsKriogen (N, О) – Признак "Использование криогенной аппаратуры"; * EvnUsluga\_Kolvo (N, О) – Количество |

* + 1. Получение информации по общей услуге

**GET api/EvnUsluga**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnUsluga\_id (N, Н) – Идентификатор оказания услуги при условии, что Класс события = «22. Оказание общей услуги» * Evn\_pid (N, Н) – Идентификатор случая-родителя (движение в КВС, посещения в ТАП) (значение сущности dbo.Evn); * Evn\_setDT (DT, Н) – Дата и время начала выполнения услуги; * UslugaComplex\_id (N, Н) – Услуга (значение справочника dbo.UslugaComplex). |
| **Ответ** | Если не передан ни один параметр, то ошибка.  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список общих услуг с информацией:   * EvnUsluga\_id (N, О) – Идентификатор оказания услуги; * Evn\_id (N, О) – Идентификатор случая оказания услуги (значение сущности dbo.Evn); * Evn\_pid (N, О) – Идентификатор случая-родителя (движение в КВС, посещения в ТАП) (значение сущности dbo.Evn); * EvnClass\_id (N, О) – Класс события (значение справочника dbo.EvnClass); * Evn\_setDT (DT, О) – Дата и время начала выполнения услуги; * Evn\_disDT (DT, О) – Дата и время окончания выполнения услуги; * UslugaPlace\_id (N, О) – Место выполнения (значение справочника dbo.UslugaPlace); * LpuSection\_id (N, У/О) – Отделение МО (значение сущности dbo.LpuSection); * Lpu\_id (N, У/О) – МО (значение справочника dbo.Lpu); * Org\_id (N, У/О) – Другая организация (значение справочника dbo.Org); * LpuSectionProfile \_id(N, У/О) – Профиль отделения МО (значение справочника dbo.LpuSectionProfile); * MedSpecOms\_id (N, У/О) – Специальность (значение справочника dbo.MedSpecOms); * MedStaffFact\_id (N, У/О) – Место работы врача, оказавшего услугу (значение сущности dbo.MedStaffFact); * PayType\_id (N, О) – Вид оплаты (значение справочника dbo.PayType); * EvnPrescr\_id (N, Н) – Назначение (значение сущности dbo.EvnPrescr); * DiagSetClass\_id (N, О) – Тип диагноза (значение справочника dbo.DiagSetClass); * Diag\_id (N, О) – Диагноз (значение справочника dbo.Diag); * UslugaCategory\_id (N, О) – Категория услуги (значение справочника dbo.UslugaCategory); * UslugaComplex\_id (N, О) – Услуга (значение справочника dbo.UslugaComplex); * EvnUsluga\_Price (N, Н) – Цена услуги; * EvnUsluga\_Kolvo (N, О) – Количество; * EvnUsluga\_Summa (N, Н) – Сумма |

* + 1. Получение информации по параклинической услуге

**GET api/EvnUslugaPar**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnUsluga\_id (N, Н) – Идентификатор оказания услуги при условии, что Класс события = «47. Параклиническая услуга»; * Person\_id (N, H) – Идентификатор человека; * Evn\_setDT (DT, Н) – Дата и время начала выполнения услуги; * UslugaComplex\_id (N, Н) – Услуга (значение справочника dbo.UslugaComplex). |
| **Ответ** | Если не передан ни один параметр, то ошибка.  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список оперативных услуг с информацией:   * EvnUsluga\_id (N, О) – Идентификатор оказания услуги; * EvnUslugaPar\_id (N, О) – Идентификатор оказания параклинической услуги; * Evn\_id (N, О) – Идентификатор случая оказания услуги (значение сущности dbo.Evn) * Evn\_setDT (DT, О) – Дата и время начала выполнения услуги; * Evn\_disDT (DT, H) – Дата и время окончания выполнения услуги; * Person\_id (N, O) – Идентификатор человека; * LpuSection\_id (N, У/О) – Отделение МО (значение сущности dbo.LpuSection). Обязательное при UslugaPlace\_id = 1. * Lpu\_id (N, У/О) – МО (значение справочника dbo.Lpu). Обязательное при UslugaPlace\_id = 2. * Org\_id (N, У/О) – Другая организация (значение справочника dbo.Org). Обязательное при UslugaPlace\_id = 3. * LpuSectionProfile \_id(N, У/О) – Профиль отделения МО (значение справочника dbo.LpuSectionProfile). Обязательное при UslugaPlace\_id = 2. * MedSpecOms\_id (N, У/О) – Специальность (значение справочника dbo.MedSpecOms). Обязательное при UslugaPlace\_id = 2. * MedStaffFact\_id (N, У/О) – Место работы врача, оказавшего услугу (значение сущности dbo.MedStaffFact). Обязательное при UslugaPlace\_id = 1. * UslugaComplex\_id (N, О) – Услуга (значение справочника dbo.UslugaComplex); * PayType\_id (N, О) – Вид оплаты (значение справочника dbo.PayType); * EvnUsluga\_Kolvo (N, О) – Количество услуг * Evn\_setDT (DT, О) – Дата и время начала выполнения услуги; * UslugaPlace\_id (N, О) – Место выполнения (значение справочника dbo.UslugaPlace); * EvnUslugaPar\_Comment (T[1024], H) – Комментарий |

* + 1. Получение информации об услуге по типу услуги

**Get api/EvnUslugaPar**

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,H) –идентификатор МО * UslugaComplexAttributeType\_id (N,O) - идентификатор типа атрибута услуги. |
| **Ответ** | В ответе возвращается массив услуг со следующими параметрами:   * PayType\_id (N,O) Тип оплаты; * Org\_uid (N,O) – идентификатор организации * MedPersonal\_id (N,O) идентификатор места работы врача * UslugaPlace\_id (N,O) Место оказания услуги * EvnUsluga\_Kolvo(N,O) Кол-во оказанных услуг * UslugaComplex\_id (N,O) идентификатор комплексной услуги * MedPersonal\_sid bigint врач-пользователь, подписавший назначение * EvnUsluga\_Result (T,H) Результат * EvnDirection\_id (N,O) –идентификатор направления * Lpu\_did (N,O) –направившее МО * Org\_did (N,O) направившая организация * MedPersonal\_did (N,Н) – идентификатор направившей организации * EvnDirection\_Num (T,H) Номер направления * EvnDirection\_setDT (D,Н)Дата направления * Person\_id (N,O)идентификатор человека в Person |
| **Пример** |  |

* + 1. Получение информации об услуге по идентификатору направления

**GET api/EvnDirectionUslugaComplex**

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,H) –идентификатор МО * EvnDirection\_id (N,O) – идентификатор направления |
| **Ответ** | В ответе возвращается массив услуг со следующими параметрами:   * UslugaComplex\_id (N,O) – идентификатор услуги * UslugaComplex\_Code (N,O) – код услуги |
| **Пример** |  |

* + 1. Получение информации об услуге по идентификатору направления

**GET api/EvnDirectionUslugaComplex**

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,H) –идентификатор МО * EvnDirection\_id (N,O) – идентификатор направления |
| **Ответ** | В ответе возвращается массив услуг со следующими параметрами:   * UslugaComplex\_id (N,O) – идентификатор услуги * UslugaComplex\_Code (N,O) – код услуги |
| **Пример** |  |

* + 1. Создание оперативной услуги

**POST api/EvnUslugaOper**

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_pid (N, О) – Идентификатор случая-родителя (движение в КВС, посещения в ТАП) (значение сущности dbo.Evn); * EvnClass\_id (N, О) – Класс события (значение справочника dbo.EvnClass). Всегда значение «43. Оперативная услуга»; * Evn\_setDT (DT, О) – Дата и время начала выполнения услуги; * Evn\_disDT (DT, О) – Дата и время окончания выполнения услуги; * UslugaPlace\_id (N, О) – Место выполнения (значение справочника dbo.UslugaPlace); * LpuSection\_id (N, У/О) – Отделение МО (значение сущности dbo.LpuSection); * Lpu\_id (N, У/О) – МО (значение справочника dbo.Lpu); * Org\_id (N, У/О) – Другая организация (значение справочника dbo.Org); * LpuSectionProfile \_id(N, У/О) – Профиль отделения МО (значение справочника dbo.LpuSectionProfile); * MedSpecOms\_id (N, У/О) – Специальность (значение справочника dbo.MedSpecOms); * MedStaffFact\_id (N, У/О) – Место работы врача, оказавшего услугу (значение сущности dbo.MedStaffFact); * PayType\_id (N, О) – Вид оплаты (значение справочника dbo.PayType); * EvnPrescr\_id (N, Н) – Назначение (значение сущности dbo.EvnPrescr); * DiagSetClass\_id (N, О) – Тип диагноза (значение справочника dbo.DiagSetClass); * Diag\_id (N, О) – Диагноз (значение справочника dbo.Diag); * UslugaComplex\_id (N, О) – Услуга (значение справочника dbo.UslugaComplex); * EvnUsluga\_Price (N, Н) – Цена услуги; * OperType\_id (N, О) – Тип операции (значение справочника dbo.OperType); * OperDiff\_id (N, О) – Категория сложности (значение справочника dbo.OperDiff); * TreatmentConditionsType\_id (N, Н) – Условие лечения (значение справочника dbo.TreatmentConditionsType); * EvnUslugaOper\_IsVMT (N, О) – Признак "Применение ВМТ"; * EvnUslugaOper\_IsMicrSurg (N, О) – Признак "Микрохирургическая"; * EvnUslugaOper\_IsOpenHeart (N, О) – Признак "На открытом сердце"; * EvnUslugaOper\_IsArtCirc (N, О) – Признак "С искусственным кровообращением"; * EvnUslugaOper\_IsEndoskop (N, О) – Признак "Использование эндоскопической аппаратуры"; * EvnUslugaOper\_IsLazer (N, О) – Признак "Использование лазерной аппаратуры"; * EvnUslugaOper\_IsKriogen (N, О) – Признак "Использование криогенной аппаратуры"; * EvnUsluga\_Kolvo (N, О) – Количество |
| **Ответ** | Создается оперативная услуга и формируется ответ:   * EvnUsluga\_id (N, О) – Идентификатор оказания услуги; * EvnUslugaOper\_id (N, О) – Идентификатор оказания оперативной услуги; * Evn\_id (N, О) – Идентификатор случая оказания услуги (значение сущности dbo.Evn) |

* + 1. Создание общей услуги

**POST api/EvnUsluga**

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_pid (N, О) – Идентификатор случая-родителя (движение в КВС, посещения в ТАП) (значение сущности dbo.Evn); * EvnClass\_id (N, О) – Класс события (значение справочника dbo.EvnClass). Всегда значение «22. Оказание общей услуги»; * Evn\_setDT (DT, О) – Дата и время начала выполнения услуги; * Evn\_disDT (DT, О) – Дата и время окончания выполнения услуги; * UslugaPlace\_id (N, О) – Место выполнения (значение справочника dbo.UslugaPlace); * LpuSection\_id (N, У/О) – Отделение МО (значение сущности dbo.LpuSection); * Lpu\_id (N, У/О) – МО (значение справочника dbo.Lpu); * Org\_id (N, У/О) – Другая организация (значение справочника dbo.Org); * LpuSectionProfile \_id(N, У/О) – Профиль отделения МО (значение справочника dbo.LpuSectionProfile); * MedSpecOms\_id (N, У/О) – Специальность (значение справочника dbo.MedSpecOms); * MedStaffFact\_id (N, У/О) – Место работы врача, оказавшего услугу (значение сущности dbo.MedStaffFact); * PayType\_id (N, О) – Вид оплаты (значение справочника dbo.PayType); * EvnPrescr\_id (N, Н) – Назначение (значение сущности dbo.EvnPrescr); * DiagSetClass\_id (N, О) – Тип диагноза (значение справочника dbo.DiagSetClass); * Diag\_id (N, О) – Диагноз (значение справочника dbo.Diag); * UslugaComplex\_id (N, О) – Услуга (значение справочника dbo.UslugaComplex); * EvnUsluga\_Price (N, Н) – Цена услуги; * EvnUsluga\_Kolvo (N, О) – Количество; * EvnUsluga\_Summa (N, Н) – Сумма |
| **Ответ** | Успешный ответ – информация по общей услуге:   * EvnUsluga\_id (N, О) – Идентификатор оказания услуги; * Evn\_id (N, О) – Идентификатор случая оказания услуги (значение сущности dbo.Evn) |

* + 1. Создание параклинической услуги

**POST api/EvnUslugaPar**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnClass\_id (N, О) – Класс события (значение справочника dbo.EvnClass). Всегда значение «47. Параклиническая услуга»; * Evn\_setDT (DT, О) – Дата и время начала выполнения услуги; * Evn\_disDT (DT, H) – Дата и время окончания выполнения услуги; * Person\_id (N, O) – Идентификатор человека; * LpuSection\_id (N, У/О) – Отделение МО (значение сущности dbo.LpuSection). Обязательное при UslugaPlace\_id = 1. * Lpu\_id (N, У/О) – МО (значение справочника dbo.Lpu). Обязательное при UslugaPlace\_id = 2. * Org\_id (N, У/О) – Другая организация (значение справочника dbo.Org). Обязательное при UslugaPlace\_id = 3. * LpuSectionProfile \_id(N, У/О) – Профиль отделения МО (значение справочника dbo.LpuSectionProfile). Обязательное при UslugaPlace\_id = 2. * MedSpecOms\_id (N, У/О) – Специальность (значение справочника dbo.MedSpecOms). Обязательное при UslugaPlace\_id = 2. * MedStaffFact\_id (N, У/О) – Место работы врача, оказавшего услугу (значение сущности dbo.MedStaffFact). Обязательное при UslugaPlace\_id = 1. * UslugaComplex\_id (N, О) – Услуга (значение справочника dbo.UslugaComplex); * PayType\_id (N, О) – Вид оплаты (значение справочника dbo.PayType); * EvnUsluga\_Kolvo (N, О) – Количество услуг * UslugaPlace\_id (N, О) – Место выполнения (значение справочника dbo.UslugaPlace). * EvnUslugaPar\_Comment (T[1024], H) – Комментарий |
| **Ответ** | Создается параклиническая услуга и формируется ответ:   * EvnUsluga\_id (N, О) – Идентификатор оказания услуги; * EvnUslugaPar\_id (N, О) – Идентификатор оказания параклинической услуги; * Evn\_id (N, О) – Идентификатор случая оказания услуги (значение сущности dbo.Evn) |

* + 1. Редактирование оперативной услуги

**PUT api/EvnUslugaOper**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnUsluga\_id (N, О) – Идентификатор оказания услуги; * Evn\_pid (N, Н) – Идентификатор случая-родителя (движение в КВС, посещения в ТАП) (значение сущности dbo.Evn); * EvnClass\_id (N, Н) – Класс события (значение справочника dbo.EvnClass). Всегда значение «43. Оперативная услуга»; * Evn\_setDT (DT, Н) – Дата и время начала выполнения услуги; * Evn\_disDT (DT, Н) – Дата и время окончания выполнения услуги; * UslugaPlace\_id (N, Н) – Место выполнения (значение справочника dbo.UslugaPlace); * LpuSection\_id (N, Н) – Отделение МО (значение сущности dbo.LpuSection); * Lpu\_id (N, Н) – МО (значение справочника dbo.Lpu); * Org\_id (N, Н) – Другая организация (значение справочника dbo.Org); * LpuSectionProfile \_id(N, Н) – Профиль отделения МО (значение справочника dbo.LpuSectionProfile); * MedSpecOms\_id (N, Н) – Специальность (значение справочника dbo.MedSpecOms); * MedStaffFact\_id (N, Н) – Место работы врача, оказавшего услугу (значение сущности dbo.MedStaffFact); * PayType\_id (N, Н) – Вид оплаты (значение справочника dbo.PayType); * EvnPrescr\_id (N, Н) – Назначение (значение сущности dbo.EvnPrescr); * DiagSetClass\_id (N, Н) – Тип диагноза (значение справочника dbo.DiagSetClass); * Diag\_id (N, Н) – Диагноз (значение справочника dbo.Diag); * UslugaComplex\_id (N, Н) – Услуга (значение справочника dbo.UslugaComplex); * EvnUsluga\_Price (N, Н) – Цена услуги; * OperType\_id (N, Н) – Тип операции (значение справочника dbo.OperType); * OperDiff\_id (N, Н) – Категория сложности (значение справочника dbo.OperDiff); * TreatmentConditionsType\_id (N, Н) – Условие лечения (значение справочника dbo.TreatmentConditionsType); * EvnUslugaOper\_IsVMT (N, Н) – Признак "Применение ВМТ"; * EvnUslugaOper\_IsMicrSurg (N, Н) – Признак "Микрохирургическая"; * EvnUslugaOper\_IsOpenHeart (N, Н) – Признак "На открытом сердце"; * EvnUslugaOper\_IsArtCirc (N, Н) – Признак "С искусственным кровообращением"; * EvnUslugaOper\_IsEndoskop (N, Н) – Признак "Использование эндоскопической аппаратуры"; * EvnUslugaOper\_IsLazer (N, Н) – Признак "Использование лазерной аппаратуры"; * EvnUslugaOper\_IsKriogen (N, Н) – Признак "Использование криогенной аппаратуры"; * EvnUsluga\_Kolvo (N, Н) – Количество |
| **Ответ** | Если случай оказания оперативной услуги найден и отредактирован, то формируется успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Редактирование общей услуги

**PUT api/EvnUsluga**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnUsluga\_id (N, О) – Идентификатор оказания услуги; * Evn\_pid (N, Н) – Идентификатор случая-родителя (движение в КВС, посещения в ТАП) (значение сущности dbo.Evn); * EvnClass\_id (N, Н) – Класс события (значение справочника dbo.EvnClass). Всегда значение «22. Оказание общей услуги»; * Evn\_setDT (DT, Н) – Дата и время начала выполнения услуги; * Evn\_disDT (DT, Н) – Дата и время окончания выполнения услуги; * UslugaPlace\_id (N, Н) – Место выполнения (значение справочника dbo.UslugaPlace); * LpuSection\_id (N, Н) – Отделение МО (значение сущности dbo.LpuSection); * Lpu\_id (N, Н) – МО (значение справочника dbo.Lpu); * Org\_id (N, Н) – Другая организация (значение справочника dbo.Org); * LpuSectionProfile \_id(N, Н) – Профиль отделения МО (значение справочника dbo.LpuSectionProfile); * MedSpecOms\_id (N, Н) – Специальность (значение справочника dbo.MedSpecOms); * MedStaffFact\_id (N, Н) – Место работы врача, оказавшего услугу (значение сущности dbo.MedStaffFact); * PayType\_id (N, Н) – Вид оплаты (значение справочника dbo.PayType); * EvnPrescr\_id (N, Н) – Назначение (значение сущности dbo.EvnPrescr); * DiagSetClass\_id (N, Н) – Тип диагноза (значение справочника dbo.DiagSetClass); * Diag\_id (N, Н) – Диагноз (значение справочника dbo.Diag); * UslugaComplex\_id (N, Н) – Услуга (значение справочника dbo.UslugaComplex); * EvnUsluga\_Price (N, Н) – Цена услуги; * EvnUsluga\_Kolvo (N, Н) – Количество; * EvnUsluga\_Summa (N, Н) – Сумма |
| **Ответ** | Если случай оказания общей услуги найден и отредактирован, то формируется успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Редактирование параклинической услуги

**PUT api/EvnUslugaPar**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnUsluga\_id (N, О) – Идентификатор оказания услуги; * Evn\_setDT (DT, H) – Дата и время начала выполнения услуги; * Evn\_disDT (DT, H) – Дата и время окончания выполнения услуги; * Person\_id (N, H) – Идентификатор человека; * LpuSection\_id (N, H) – Отделение МО (значение сущности dbo.LpuSection); * Lpu\_id (N, H) – МО (значение справочника dbo.Lpu); * Org\_id (N, H) – Другая организация (значение справочника dbo.Org); * LpuSectionProfile \_id(N, H) – Профиль отделения МО (значение справочника dbo.LpuSectionProfile); * MedSpecOms\_id (N, H) – Специальность (значение справочника dbo.MedSpecOms); * MedStaffFact\_id (N, H) – Место работы врача, оказавшего услугу (значение сущности dbo.MedStaffFact); * UslugaComplex\_id (N, H) – Услуга (значение справочника dbo.UslugaComplex); * PayType\_id (N, H) – Вид оплаты (значение справочника dbo.PayType); * EvnUsluga\_Kolvo (N, H) – Количество услуг * Evn\_setDT (DT, H) – Дата и время начала выполнения услуги; * UslugaPlace\_id (N, H) – Место выполнения (значение справочника dbo.UslugaPlace); * EvnUslugaPar\_Comment (T[1024], H) – Комментарий |
| **Ответ** | Если случай оказания параклинической услуги найден и отредактирован, то формируется успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Получение списка анестезии по услуге

**GET api/EvnUslugaOperAnestList**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnUslugaOper\_id (N, О) – Идентификатор оказания оперативной услуги (значение сущности dbo.EvnUslugaOper) |
| **Ответ** | Успешный ответ – список анестезии по услуге:   * EvnUslugaOperAnest\_id (N, О) – Идентификатор использования анестезии; * EvnUslugaOper\_id (N, О) – Идентификатор оказания оперативной услуги (значение сущности dbo.EvnUslugaOper); * AnesthesiaClass\_id (N, О) – Тип анестезии (значение справочника dbo.AnesthesiaClass) |

* + 1. Получение информации по анестезии

**GET api/EvnUslugaOperAnest**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnUslugaOperAnest\_id (N, Н) – Идентификатор использования анестезии; * EvnUslugaOper\_id (N, Н) – Идентификатор оказания оперативной услуги (значение сущности dbo.EvnUslugaOper); * AnesthesiaClass\_id (N, Н) – Тип анестезии (значение справочника dbo.AnesthesiaClass) |
| **Ответ** | Если не передан ни один параметр, то ошибка.  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список анестезий с информацией:   * EvnUslugaOperAnest\_id (N, О) – Идентификатор использования анестезии; * EvnUslugaOper\_id (N, О) – Идентификатор оказания оперативной услуги (значение сущности dbo.EvnUslugaOper); * AnesthesiaClass\_id (N, О) – Тип анестезии (значение справочника dbo.AnesthesiaClass) |

* + 1. Создание анестезии на оперативной услуге

**POST api/EvnUslugaOperAnest**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnUslugaOper\_id (N, О) – Идентификатор оказания оперативной услуги (значение сущности dbo.EvnUslugaOper); * AnesthesiaClass\_id (N, О) – Тип анестезии (значение справочника dbo.AnesthesiaClass) |
| **Ответ** | Успешный ответ – список анестезии по услуге:   * EvnUslugaOperAnest\_id (N, О) – Идентификатор использования анестезии |

* + 1. Редактирование анестезии на оперативной услуге

**PUT api/EvnUslugaOperAnest**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnUslugaOperAnest\_id (N, О) – Идентификатор использования анестезии; * EvnUslugaOper\_id (N, Н) – Идентификатор оказания оперативной услуги (значение сущности dbo.EvnUslugaOper); * AnesthesiaClass\_id (N, Н) – Тип анестезии (значение справочника dbo.AnesthesiaClass) |
| **Ответ** | Если случай использования анестезии найден и отредактирован, то формируется успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Получение списка осложнений на услуге

**GET api/EvnAggList**

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_id – Идентификатор случая услуги-родителя (значение сущности dbo.Evn) |
| **Ответ** | Успешный ответ – список осложнений по услуге:   * EvnAgg\_id (N, О) – Идентификатор осложнения; * Evn\_id (N, О) – Идентификатор услуги-родителя (значение сущности dbo.Evn); * AggType\_id (N, О) – Тип осложнения (значение справочника dbo.AggType); * AggWhen\_id (N, О) – Период, в котором произошло осложнение (значение справочника dbo.AggWhen) |

* + 1. Получение информации по осложнению услуги

**GET api/EvnAgg**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnAgg\_id (N, Н) – Идентификатор осложнения; * Evn\_id (N, Н) – Идентификатор услуги-родителя (значение сущности dbo.Evn); * AggType\_id (N, Н) – Тип осложнения (значение справочника dbo.AggType) |
| **Ответ** | Если не передан ни один параметр, то ошибка.  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список осложнений с информацией:   * EvnAgg\_id (N, О) – Идентификатор осложнения; * Evn\_id (N, О) – Идентификатор услуги-родителя (значение сущности dbo.Evn); * AggType\_id (N, О) – Тип осложнения (значение справочника dbo.AggType); * AggWhen\_id (N, О) – Период, в котором произошло осложнение (значение справочника dbo.AggWhen) |

* + 1. Создание осложнения на услуге

**POST api/EvnAgg**

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_id (N, О) – Идентификатор услуги-родителя (значение сущности dbo.Evn); * AggType\_id (N, О) – Тип осложнения (значение справочника dbo.AggType); * AggWhen\_id (N, О) – Период, в котором произошло осложнение (значение справочника dbo.AggWhen) |
| **Ответ** | Создается осложнение на услуге и формируется ответ:   * EvnAgg\_id (N, О) – Идентификатор осложнения |

* + 1. Редактирование осложнения на услуге

**PUT api/EvnAgg**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnAgg\_id (N, О) – Идентификатор осложнения; * Evn\_id (N, Н) – Идентификатор услуги-родителя (значение сущности dbo.Evn); * AggType\_id (N, Н) – Тип осложнения (значение справочника dbo.AggType); * AggWhen\_id (N, Н) – Период, в котором произошло осложнение (значение справочника dbo.AggWhen) |
| **Ответ** | Если осложнение найдено и отредактировано, то формируется успешный ответ – пустой ответ с кодом ошибки «0» |

* 1. Описание методов работы с документами нетрудоспособности
  2. 1. Получение списка документов нетрудоспособности по случаю лечения

**GET api/StickList**

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_pid (N, О) – Идентификатор события-родителя (ТАП, КВС) |
| **Ответ** | Успешный ответ – список ЛВН с указанием следующих параметров:   * EvnStickBase\_id (N, О) – Идентификатор ЛВН; * Evn\_id (N, О) – Идентификатор события выписки документа нетрудоспособности; * StickType\_id (N, О) – Тип документа нетрудоспособности (значение справочника dbo.StickType); * EvnStick\_Num (S, О) – Номер; * EvnStick\_SetDate (D, О) – Дата выдачи; |

* + 1. Получение информации по ЛВН

**GET api/Stick**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnStickBase\_id (N, Н) – Идентификатор документа нетрудоспособности; * Evn\_pid (N, Н) – Идентификатор события-родителя (ТАП, КВС); * EvnStick\_Num (N, Н) – Номер; * EvnStick\_SetDate (D, Н) – Дата выдачи ЛВН |
| **Ответ** | Если не передан ни один параметр, то ошибка.  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список ЛВН с информацией:   * EvnStickBase\_id (N, О) – Идентификатор ЛВН; * Evn\_id (N, О) – Идентификатор события выписки документа нетрудоспособности; * Evn\_pid (N, О) – Идентификатор события-родителя (ТАП, КВС); * StickType\_id (N, О) – Тип документа нетрудоспособности (значение справочника dbo.StickType); * EvnStick\_isOriginal (N, О) – Тип документа (1 – Оригинал, 2 - Дубликат); * StickWorkType\_id (N, О) – Тип занятости (значение справочника dbo.StickWorkType); * StickOrder\_id (N, О) – Порядок выдачи (значение справочника dbo.StickOrder); * EvnStick\_prid (N, Н) – Предыдущий ЛВН (значение сущности dbo.EvnStick); * EvnStick\_Num (N, О) – Номер; * EvnStick\_SetDate (D, О) – Дата выдачи ЛВН; * StickCause\_id (N, О) – Причина нетрудоспособности (значение справочника dbo.StickCause); * StickCauseDopType\_id (N, Н) – Дополнительный код нетрудоспособности (значение справочника dbo.StickCauseDopType); * StickCause\_did (N, Н) – Код изменения нетрудоспособности (значение справочника dbo.StickCause); * StickIrregularity\_id (N, Н) – Код нарушения режима (значение справочника dbo.StickIrregularity); * EvnStick\_StacBegDate (D, Н) – Дата начала лечения в стационаре; * EvnStick\_StacEndDate (D, Н) – Дата окончания лечения в стационаре; * EvnStick\_mseDate (D, Н) – Дата направления в бюро МСЭ; * EvnStick\_mseRegDate (D, Н) – Дата регистрации документов в бюро МСЭ; * EvnStick\_mseExamDate (D, Н) – Дата освидетельствования документов в бюро МСЭ; * InvalidGroupType\_id (N, Н) – Установленная группа инвалидности (значение справочника dbo.InvalidGroupType); * StickLeaveType\_id (N, Н) – Исход ЛВН (значение справочника dbo.StickLeaveType); * EvnStick\_DisDate (D, Н) – Дата исхода; * MedStaffFact\_id (N, Н) – Место работы врача (значение справочника dbo.MedStaffFact); * Lpu\_oid (N, Н) – Направлен в другое МО (значение сущности dbo.Lpu) |

* + 1. Получение информации по Справке учащегося

**GET api/StickStudent**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnStickBase\_id (N, Н) – Идентификатор документа нетрудоспособности; * Evn\_pid (N, Н) – Идентификатор события-родителя (ТАП, КВС); * EvnStick\_Num (N, Н) – Номер; * EvnStick\_SetDate (D, Н) – Дата выдачи |
| **Ответ** | Если не передан ни один параметр, то ошибка.  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список справок учащегося с информацией:   * EvnStickBase\_id (N, О) – Идентификатор; * Evn\_id (N, О) – Идентификатор события выписки документа нетрудоспособности; * Evn\_pid (N, О) – Идентификатор события-родителя (ТАП, КВС); * StickType\_id (N, О) – Тип документа нетрудоспособности (значение справочника dbo.StickType); * EvnStick\_Num (N, О) – Номер; * EvnStick\_SetDate (D, О) – Дата выдачи; * Org\_id (N, О) – Организация (для которой оформляется справка) (значение справочника dbo.Org); * StickRecipient\_id (N, О) – Получатель справки (значение справочника dbo.StickRecipient); * StickCause\_id (N, О) – Причина нетрудоспособности (значение справочника dbo.StickCause); * EvnStick\_idContact (N, О) - Наличие контакта с инфекционными больными. * EvnStick\_ContactDescr (String, Н) - Описание контакта; * MedStaffFact\_id (N, О) – Место работы врача (значение сущности dbo.MedStaffFact) |

* + 1. Создание ЛВН

**POST api/Stick**

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_pid (N, О) – Идентификатор события-родителя; * StickType\_id (N, О) – Тип документа нетрудоспособности (значение справочника dbo.StickType). Всегда «1. ЛВН»; * EvnStick\_isOriginal (N, О) – Тип документа (1 – Оригинал, 2 - Дубликат); * StickWorkType\_id (N, О) – Тип занятости (значение справочника dbo.StickWorkType); * StickOrder\_id (N, О) – Порядок выдачи (значение справочника dbo.StickOrder); * EvnStick\_prid (N, Н) – Предыдущий ЛВН (значение сущности dbo.EvnStick); * EvnStick\_Num (N, О) – Номер; * EvnStick\_SetDate (D, О) – Дата выдачи ЛВН; * StickCause\_id (N, О) – Причина нетрудоспособности (значение справочника dbo.StickCause); * StickCauseDopType\_id (N, Н) – Дополнительный код нетрудоспособности (значение справочника dbo.StickCauseDopType); * StickCause\_did (N, Н) – Код изменения нетрудоспособности (значение справочника dbo.StickCause); * StickIrregularity\_id (N, Н) – Код нарушения режима (значение справочника dbo.StickIrregularity); * EvnStick\_StacBegDate (D, Н) – Дата начала лечения в стационаре; * EvnStick\_StacEndDate (D, Н) – Дата окончания лечения в стационаре; * EvnStick\_mseDate (D, Н) – Дата направления в бюро МСЭ; * EvnStick\_mseRegDate (D, Н) – Дата регистрации документов в бюро МСЭ; * EvnStick\_mseExamDate (D, Н) – Дата освидетельствования документов в бюро МСЭ; * InvalidGroupType\_id (N, Н) – Установленная группа инвалидности (значение справочника dbo.InvalidGroupType); * StickLeaveType\_id (N, Н) – Исход ЛВН (значение справочника dbo.StickLeaveType); * EvnStick\_DisDate (D, Н) – Дата исхода; * MedStaffFact\_id (N, Н) – Место работы врача (значение справочника dbo.MedStaffFact); * Lpu\_oid (N, Н) – Направлен в другое МО (значение сущности dbo.Lpu) |
| **Ответ** | Успешный ответ – информация по ЛВН:   * EvnStickBase\_id (N, О) – Идентификатор ЛВН; * Evn\_id (N, О) – Идентификатор события выписки документа нетрудоспособности |

* + 1. Создание справки учащегося

**POST api/StickStudent**

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_pid (N, О) – Идентификатор события-родителя; * StickType\_id (N, О) – Тип документа нетрудоспособности (значение справочника dbo.StickType). Всегда «2. Справка учащегося»; * EvnStick\_Num (N, О) – Номер; * EvnStick\_SetDate (D, О) – Дата выдачи; * Org\_id (N, О) – Организация (для которой оформляется справка) (значение справочника dbo.Org); * StickRecipient\_id (N, О) – Получатель справки (значение справочника dbo.StickRecipient); * StickCause\_id (N, О) – Причина нетрудоспособности (значение справочника dbo.StickCause); * EvnStick\_idContact (N, О) - Наличие контакта с инфекционными больными. (0 - Нет, 1 - Да); * EvnStick\_ContactDescr (String, Н) - Описание контакта; * MedStaffFact\_id (N, О) – Место работы врача (значение сущности dbo.MedStaffFact) |
| **Ответ** | Успешный ответ – информация по Справке учащегося:   * EvnStickBase\_id (N, О) – Идентификатор ЛВН; * Evn\_id (N, О) – Идентификатор события выписки документа нетрудоспособности |

* + 1. Редактирование информации по ЛВН

**PUT api/ Stick**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnStickBase\_id (N, О) – Идентификатор ЛВН; * Evn\_pid (N, Н) – Идентификатор события-родителя; * EvnStick\_isOriginal (N, Н) – Тип документа (1 – Оригинал, 2 - Дубликат); * StickWorkType\_id (N, Н) – Тип занятости (значение справочника dbo.StickWorkType); * StickOrder\_id (N, Н) – Порядок выдачи (значение справочника dbo.StickOrder); * EvnStick\_prid (N, Н) – Предыдущий ЛВН (значение сущности dbo.EvnStick); * EvnStick\_Num (N, Н) – Номер; * EvnStick\_SetDate (D, Н) – Дата выдачи ЛВН; * StickCause\_id (N, Н) – Причина нетрудоспособности (значение справочника dbo.StickCause); * StickCauseDopType\_id (N, Н) – Дополнительный код нетрудоспособности (значение справочника dbo.StickCauseDopType); * StickCause\_did (N, Н) – Код изменения нетрудоспособности (значение справочника dbo.StickCause); * StickIrregularity\_id (N, Н) – Код нарушения режима (значение справочника dbo.StickIrregularity); * EvnStick\_StacBegDate (D, Н) – Дата начала лечения в стационаре; * EvnStick\_StacEndDate (D, Н) – Дата окончания лечения в стационаре; * EvnStick\_mseDate (D, Н) – Дата направления в бюро МСЭ; * EvnStick\_mseRegDate (D, Н) – Дата регистрации документов в бюро МСЭ; * EvnStick\_mseExamDate (D, Н) – Дата освидетельствования документов в бюро МСЭ; * InvalidGroupType\_id (N, Н) – Установленная группа инвалидности (значение справочника dbo.InvalidGroupType); * StickLeaveType\_id (N, Н) – Исход ЛВН (значение справочника dbo.StickLeaveType); * EvnStick\_DisDate (D, Н) – Дата исхода; * MedStaffFact\_id (N, Н) – Место работы врача (значение справочника dbo.MedStaffFact); * Lpu\_oid (N, Н) – Направлен в другое МО (значение сущности dbo.Lpu) |
| **Ответ** | * Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Редактирование информации по справке учащегося

**PUT api/StickStudent**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnStickBase\_id (N, О) – Идентификатор; * Evn\_pid (N, О) – Идентификатор события-родителя; * EvnStick\_Num (N, Н) – Номер; * EvnStick\_SetDate (D, Н) – Дата выдачи; * Org\_id (N, Н) – Организация (для которой оформляется справка) (значение справочника dbo.Org); * StickRecipient\_id (N, Н) – Получатель справки (значение справочника dbo.StickRecipient); * StickCause\_id (N, Н) – Причина нетрудоспособности (значение справочника dbo.StickCause); * EvnStick\_idContact (N, Н) - Наличие контакта с инфекционными больными * EvnStick\_ContactDescr (String, Н) - Описание контакта; * MedStaffFact\_id (N, Н) – Место работы врача (значение сущности dbo.MedStaffFact) |
| **Ответ** | * Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Получение списка периодов освобождения от работы

**GET api/WorkReleaseList**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnStickBase\_id (N, О) – Идентификатор документа нетрудоспособности. |
| **Ответ** | Успешный ответ – список периодов освобождения от работы с указанием следующих параметров:   * EvnStickBase\_id (N, О) – Идентификатор документа нетрудоспособности; * EvnStickWorkRelease\_id (N, О) – идентификатор периода освобождения от работы; * EvnStickWorkRelease\_begDate (D, О) – Дата начала освобождения от работы; * EvnStickWorkRelease\_endDate (D, О) – Дата окончания освобождения от работы |

* + 1. Получение информации по периоду освобождения от работы

**GET api/WorkRelease**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnStickWorkRelease\_id (N, Н) – идентификатор периода освобождения от работы; * EvnStickBase\_id (N, Н) – Идентификатор документа нетрудоспособности; * EvnStickWorkRelease\_begDate (D, Н) – Дата начала освобождения от работы; * EvnStickWorkRelease\_endDate (D, Н) – Дата окончания освобождения от работы |
| **Ответ** | Если не передан ни один параметр, то ошибка.  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список периодов освобождений от работы с информацией:   * EvnStickBase\_id (N, О) – Идентификатор документа нетрудоспособности; * EvnStickWorkRelease\_id (N, О) – идентификатор периода освобождения от работы; * EvnStickWorkRelease\_isDraft (N, Н) - Признак «Черновик за другую МО»; * EvnStickWorkRelease\_begDate (D, О) – Дата начала освобождения от работы; * EvnStickWorkRelease\_endDate (D, О) – Дата окончания освобождения от работы; * LpuSection\_id (N, О) – Отделение МО врача 1 (значение сущности dbo.LpuSection); * MedStaffFact\_id (N, О) – Место работы врача 1 (значение сущности dbo.MedStaffFact); * MedStaffFact2\_id (N, Н) – Место работы врача 2 (значение сущности dbo.MedStaffFact); * MedStaffFact3\_id (N, Н) – Место работы врача 3 (значение сущности dbo.MedStaffFact); * EvnStickWorkRelease\_isPredVK (N, Н) – Признак «Председатель ВК». |
| **Пример** |  |

* + 1. Создание периода освобождения от работы

**POST api/WorkRelease**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnStickBase\_id (N, О) – Идентификатор документа нетрудоспособности;   EvnStickWorkRelease\_isDraft (N, О) - Признак «Черновик за другую МО».Возможные значения: 0 и 1, - где 0 – нет, 1 – да;   * EvnStickWorkRelease\_begDate (D, О) – Дата начала освобождения от работы; * EvnStickWorkRelease\_endDate (D, О) – Дата окончания освобождения от работы; * LpuSection\_id (N, О) – Отделение МО врача 1; * MedStaffFact\_id (N, О) – Место работы врача 1; * MedStaffFact2\_id (N, Н) – Место работы врача 2; * MedStaffFact3\_id (N, Н) – Место работы врача 3;   EvnStickWorkRelease\_isPredVK (N, О) – Признак «Председатель ВК».».Возможные значения: 0 и 1, - где 0 – нет, 1 – да. |
| **Ответ** | Успешный ответ – информация по периоду освобождения от работы:   * EvnStickBase\_id (N, О) – Идентификатор документа нетрудоспособности; * EvnStickWorkRelease\_id (N, О) – идентификатор периода освобождения от работы |

* + 1. Редактирование информации по периоду освобождения от работы

**PUT api/WorkRelease**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnStickWorkRelease\_id (N, О) – идентификатор периода освобождения от работы; * EvnStickWorkRelease\_isDraft (N, Н) - Признак «Черновик за другую МО»; * EvnStickWorkRelease\_begDate (D, Н) – Дата начала освобождения от работы; * EvnStickWorkRelease\_endDate (D, Н) – Дата окончания освобождения от работы; * LpuSection\_id (N, Н) – Отделение МО врача 1; * MedStaffFact\_id (N, Н) – Место работы врача 1; * MedStaffFact2\_id (N, Н) – Место работы врача 2; * MedStaffFact3\_id (N, Н) – Место работы врача 3; * EvnStickWorkRelease\_isPredVK (N, Н) – Признак «Председатель ВК». |
| **Ответ** | * Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Получение списка связей случая лечения с ЛВН

**GET api/EvnLinkList**

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_id (N, О) – идентификатор случая лечения |
| **Ответ** | Успешный ответ – список документов случая:   * Evn\_id (N, О) – Идентификатор случая лечения**;** * Evn\_lid (N, О) – Идентификатор случая выписки ЛВН/Справки учащегося |

* + 1. Создание связи случая лечения с ЛВН

**POST api/EvnLink**

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_id (N, О) – Идентификатор случая лечения**;** * Evn\_lid (N, О) – Идентификатор случая выписки ЛВН/Справки учащегося |
| **Ответ** | * Evn\_id (N, О) – идентификатор случая лечения; * EvnXml\_id (N, О) – идентификатор связи случая лечения с ЛВН. |
| **Пример** |  |

* + 1. Удаление связи случая лечения с ЛВН

**DELETE api/EvnLink**

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_id (N, О) – Идентификатор случая лечения**;** * Evn\_lid (N, О) – Идентификатор случая выписки ЛВН/Справки учащегося |
| **Ответ** | Если связь найдена и удалена, то формируется успешный ответ – пустой ответ с кодом ошибки «0» |

* 1. Описание методов работы со свидетельствами
  2. 1. Получение списка свидетельств о рождении по человеку

**GET api/BirthSvidList**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, О) – Идентификатор человека |
| **Ответ** | Успешный ответ – список свидетельств о рождении:   * BirthSvid\_id (N, О) – Идентификатор Свидетельства о рождении; * BirthSvid\_Ser (S, О) – Серия Свидетельства о рождении; * BirthSvid\_Num (S, О) – Номер Свидетельства о рождении; * BirthSvid\_GiveDate (D, О) – Дата выдачи Свидетельства о рождении |

* + 1. Получение информации по Свидетельству о рождении

**GET api/BirthSvid**

|  |  |
| --- | --- |
| **Входящие параметры** | * BirthSvid\_id (N, О) – Идентификатор Свидетельства о рождении |
| **Ответ** | Успешный ответ – информация по Свидетельству о рождении:   * BirthSvid\_id (N, О) – Идентификатор Свидетельства о рождении; * ReceptType\_id (N, О) – Тип свидетельства (значение справочника dbo.ReceptType); * BirthSvid\_Ser (S, О) – Серия свидетельства; * BirthSvid\_Num (S, О) – Номер свидетельства; * BirthSvid\_GiveDate (D, О) – Дата выдачи свидетельства; * BirthEmployment\_id (N, О) – Занятость матери (значение справочника dbo.BirthEmployment); * BirthEducation\_id (N, О) – Образование матери (значение справочника dbo.BirthEducation); * BirthFamilyStatus\_id (N, Н) – Семейное положение матери (значение справочника dbo.BirthFamilyStatus); * LpuSection\_id (N, О) – Отделение МО (значение сущности dbo.LpuSection); * MedStaffFact\_id (N, О) – Место работы врача (значение сущности dbo.MedStaffFact); * BirthMedPersonalType\_id (N, О) – Вид мед. Персонала, принимающего роды (значение справочника dbo.BirthMedPersonalType); * OrgHead\_id (N, О) – Руководитель МО (значение сущности dbo.OrgHead); * LpuLicence\_id (S, О) – Лицензия МО (значение сущности dbo.LpuLicence); * BirthSvid\_BirthDT\_Date (DT, О) – Дата и время родов; * BirthPlace\_id (N, О) – Место родов (значение справочника dbo.BirthPlace); * BirthSpecialist\_id (N, О) – Специалист, принимавший роды (значение справочника dbo.BirthSpecialist); * BirthSvid\_IsMnogoplod (N, О) – Признак «Многоплодные роды» (0 – Нет, 1 - Да); * BirthSvid\_PlodIndex (N, Н) – Который по счету; * BirthSvid\_PlodCount (N, Н) – Всего плодов; * BirthChildResult\_id (N, О) – Результат родов (значение справочника dbo.BirthChildResult); * BirthSvid\_ChildCount (N, О) – Который ребенок (по счету); * BirthSvid\_Week (S, О) – Первая явка неделя; * BirthSvid\_Mass (N, О) – Масса ребенка; * Okei\_mid (N, О) – Единица измерения массы (значение справочника dbo.Okei); * BirthSvid\_Height (N, О) – Рост ребенка в см; * Sex\_id (N, О) – Пол ребенка (значение справочника dbo.Sex); * BirthSvid\_ChildFamil (S, О) – Фамилия ребенка; * Address\_rid (N, Н) – Адрес (значение сущности dbo.Address); * Person\_rid (S, Н) – Получатель свидетельства (значение сущности dbo.Person); * BirthSvid\_RcpDocument (S, Н) – Документ получателя свидетельства (Серия, Номер, Кем выдан); * DeputyKind\_id (N, Н) – Отношение к ребенку получателя свидетельства (значение справочника dbo.DeputyKind); * BirthSvid\_RcpDate (D, О) – Дата получения свидетельсвта; * BirthSvid\_IsFromMother (N, О) – Признак «Записано со слов матери» (0 – Нет, 1 - Да) |

* + 1. Создание Свидетельства о рождении

**POST api/BirthSvid**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, О) – Идентификатор человека; * ReceptType\_id (N, О) – Тип свидетельства (значение справочника dbo.ReceptType); * BirthSvid\_Ser (S, О) – Серия свидетельства; * BirthSvid\_Num (S, О) – Номер свидетельства; * BirthSvid\_GiveDate (D, О) – Дата выдачи свидетельства; * BirthEmployment\_id (N, О) – Занятость матери (значение справочника dbo.BirthEmployment); * BirthEducation\_id (N, О) – Образование матери (значение справочника dbo.BirthEducation); * BirthFamilyStatus\_id (N, Н) – Семейное положение матери (значение справочника dbo.BirthFamilyStatus); * LpuSection\_id (N, О) – Отделение МО (значение сущности dbo.LpuSection); * MedStaffFact\_id (N, О) – Место работы врача (значение сущности dbo.MedStaffFact); * BirthMedPersonalType\_id (N, О) – Вид мед. Персонала, принимающего роды (значение справочника dbo.BirthMedPersonalType); * OrgHead\_id (N, О) – Руководитель МО (значение сущности dbo.OrgHead); * LpuLicence\_id (S, О) – Лицензия МО (значение сущности dbo.LpuLicence); * BirthSvid\_BirthDT (DT, О) – Дата и время родов; * BirthPlace\_id (N, О) – Место родов (значение справочника dbo.BirthPlace); * BirthSpecialist\_id (N, О) – Специалист, принимавший роды (значение справочника dbo.BirthSpecialist); * BirthSvid\_IsMnogoplod (N, О) – Признак «Многоплодные роды» (0 – Нет, 1 – Да); * BirthSvid\_PlodIndex (N, Н) – Который по счету; * BirthSvid\_PlodCount (N, Н) – Всего плодов; * BirthChildResult\_id (N, О) – Результат родов (значение справочника dbo.BirthChildResult); * BirthSvid\_ChildCount (N, О) – Который ребенок (по счету); * BirthSvid\_Week (S, О) – Первая явка неделя; * BirthSvid\_Mass (N, О) – Масса ребенка; * Okei\_mid (N, О) – Единица измерения массы (значение справочника dbo.Okei); * BirthSvid\_Height (N, О) – Рост ребенка в см; * Sex\_id (N, О) – Пол ребенка (значение справочника dbo.Sex); * BirthSvid\_ChildFamil (S, О) – Фамилия ребенка; * Address\_rid (N, Н) – Адрес (значение сущности dbo.Address); * Person\_rid (S, Н) – Получатель свидетельства (значение сущности dbo.Person); * BirthSvid\_RcpDocument (S, Н) – Документ получателя свидетельства (Серия, Номер, Кем выдан); * DeputyKind\_id (N, Н) – Отношение к ребенку получателя свидетельства (значение справочника dbo.DeputyKind); * BirthSvid\_RcpDate (D, О) – Дата получения свидетельсвта; * BirthSvid\_IsFromMother (N, О) – Признак «Записано со слов матери» (0 – Нет, 1 – Да) |
| **Ответ** | Производятся проверки:   1. Если по переданному значению Person\_id не найден пациент, то ошибка «Пациент не найден в системе». 2. 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 документа не производится.   1. Если Свидетельство о рождении с аналогичными передаваемыми параметрами для данного человека найдено в системе, создание документа не происходит, ошибка «Данные документа не прошли проверку на дублирование.». 2. Если в передаваемом параметре:  * ReceptType\_id, * BirthEmployment\_id * BirthEducation\_id * BirthFamilyStatus\_id * LpuSection\_id * MedStaffFact\_id * BirthMedPersonalType\_id * OrgHead\_id * LpuLicence\_id * BirthPlace\_id * BirthSpecialist\_id * BirthChildResult\_id * Sex\_id * Address\_rid * Person\_rid * DeputyKind\_id   передается значение id, не найденное в системе, ошибка: «Значения для <Наименования полей через запятую> не найдены в справочниках системы.», создание документа не происходит. Аналогичный контроль производится для параметров, по которым ожидается значение 0/1 (да/нет) (BirthSvid\_IsFromMother). Если контроль не пройден, то ошибка «Значение поля <названия полей через запятую> не соответствует ожидаемому», создание документа не производится.  Успешный ответ – информация по Свидетельству о рождении:   1. Person\_id (N, О) – Идентификатор человека; 2. BirthSvid\_id (N, О) – Идентификатор Свидетельства о рождении |

* + 1. Редактирование информации по Свидетельству о рождении

**PUT api/BirthSvid**

|  |  |
| --- | --- |
| **Входящие параметры** | * BirthSvid\_id (N, О) – Идентификатор Свидетельства о рождении; * ReceptType\_id (N, Н) – Тип свидетельства (значение справочника dbo.ReceptType); * BirthSvid\_Ser (S, Н) – Серия свидетельства; * BirthSvid\_Num (S, Н) – Номер свидетельства; * BirthSvid\_GiveDate (D, Н) – Дата выдачи свидетельства; * BirthEmployment\_id (N, Н) – Занятость матери (значение справочника dbo.BirthEmployment); * BirthEducation\_id (N, Н) – Образование матери (значение справочника dbo.BirthEducation); * BirthFamilyStatus\_id (N, Н) – Семейное положение матери (значение справочника dbo.BirthFamilyStatus); * LpuSection\_id (N, Н) – Отделение МО (значение сущности dbo.LpuSection); * MedStaffFact\_id (N, Н) – Место работы врача (значение сущности dbo.MedStaffFact); * BirthMedPersonalType\_id (N, Н) – Вид мед. Персонала, принимающего роды (значение справочника dbo.BirthMedPersonalType); * OrgHead\_id (N, Н) – Руководитель МО (значение сущности dbo.OrgHead); * LpuLicence\_id (S, Н) – Лицензия МО (значение сущности dbo.LpuLicence); * BirthSvid\_BirthDT\_Date (DT, Н) – Дата и время родов; * BirthPlace\_id (N, Н) – Место родов (значение справочника dbo.BirthPlace); * BirthSpecialist\_id (N, Н) – Специалист, принимавший роды (значение справочника dbo.BirthSpecialist); * BirthSvid\_IsMnogoplod (N, Н) – Признак «Многоплодные роды» (0 – Нет, 1 – Да); * BirthSvid\_PlodIndex (N, Н) – Который по счету; * BirthSvid\_PlodCount (N, Н) – Всего плодов; * BirthChildResult\_id (N, Н) – Результат родов (значение справочника dbo.BirthChildResult); * BirthSvid\_ChildCount (N, Н) – Который ребенок (по счету); * BirthSvid\_Week (S, Н) – Первая явка неделя; * BirthSvid\_Mass (N, Н) – Масса ребенка; * Okei\_mid (N, Н) – Единица измерения массы (значение справочника dbo.Okei); * BirthSvid\_Height (N, Н) – Рост ребенка в см; * Sex\_id (N, О) – Пол ребенка (значение справочника dbo.Sex); * BirthSvid\_ChildFamil (S, Н) – Фамилия ребенка; * Address\_rid (N, Н) – Адрес (значение сущности dbo.Address); * Person\_rid (S, Н) – Получатель свидетельства (значение сущности dbo.Person); * BirthSvid\_RcpDocument (S, Н) – Документ получателя свидетельства (Серия, Номер, Кем выдан); * DeputyKind\_id (N, Н) – Отношение к ребенку получателя свидетельства (значение справочника dbo.DeputyKind); * BirthSvid\_RcpDate (D, Н) – Дата получения свидетельсвта; * BirthSvid\_IsFromMother (N, Н) – Признак «Записано со слов матери» (0 – Нет, 1 – Да) |
| **Ответ** | Выполняются следующие проверки:   1. Если по переданному значению BirthSvid\_id не найден документ, то ошибка «Свидетельство о рождении не найдено в системе». 2. Если в любом передаваемом параметре передается значение, не соответствующее ожидаемому формату, например:  * ожидается числовое значение, а передано текстовое значение; * ожидается параметр, состоящий из 11 символов, а передан параметр, состоящий из большего или меньшего количества символов;   то, ошибка: «Неверный формат поля <Наименования полей через запятую>», обновление документа не производится.   1. Если в передаваемом параметре:  * ReceptType\_id, * BirthEmployment\_id * BirthEducation\_id * BirthFamilyStatus\_id * LpuSection\_id * MedStaffFact\_id * BirthMedPersonalType\_id * OrgHead\_id * LpuLicence\_id * BirthPlace\_id * BirthSpecialist\_id * BirthChildResult\_id * Sex\_id * Address\_rid * Person\_rid * DeputyKind\_id   передается значение id, не найденное в системе, ошибка: «Значения для <Наименования полей через запятую> не найдены в справочниках системы.», обновление данных документа не происходит. Аналогичный контроль производится для параметров, по которым ожидается значение 0/1 (да/нет) (BirthSvid\_IsFromMother). Если контроль не пройден, то ошибка «Значение поля <названия полей через запятую> не соответствует ожидаемому», обновление данных документа не производится.  Успешный ответ – пустой ответ с кодом ошибки «0» |
| **Пример** |  |

* + 1. Получение списка свидетельств о смерти по человеку

**GET api/DeathSvidList**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, О) – Идентификатор человека |
| **Ответ** | Успешный ответ – список свидетельств о смерти:   * DeathSvid\_id (N, О) – Идентификатор Свидетельства о смерти; * DeathSvid\_Ser (S, О) – Серия Свидетельства о смерти; * DeathSvid\_Num (S, О) – Номер Свидетельства о смерти; * DeathSvid\_GiveDate (D, О) – Дата выдачи Свидетельства о смерти |

* + 1. Получение информации по Свидетельству о смерти

**GET api/DeathSvid**

|  |  |
| --- | --- |
| **Входящие параметры** | * DeathSvid\_id (N, О) – Идентификатор Свидетельства о смерти |
| **Ответ** | Успешный ответ – информация по Свидетельству о смерти:   * DeathSvid\_id (N, О) – Идентификатор Свидетельства о смерти; * ReceptType\_id (N, О) - Тип свидетельства (значение справочника dbo.ReceptType); * DeathSvid\_Ser (S, О) – Серия свидетельства; * DeathSvid\_Num (N, О) – Номер свидетельства; * DeathSvid\_GiveDate (D, О) – Дата выдачи свидетельства; * DeathSvidType\_id (N, О) – Вид свидетельства (значение справочника dbo.DeathSvidType); * DeathSvid\_OldSer (S, Н) – Серия предыдущего свидетельства; * DeathSvid\_OldNum (N, Н) – Номер предыдущего свидетельства; * DeathSvid\_OldGiveDate (D, Н) – Дата выдачи предыдущего свидетельства; * DeathSvid\_BirthDateStr (D, Н) – Дата рождения; * DeathSvid\_isBirthDate (N, Н) – Признак "Неполная/неизвестная дата рождения"; * DeathSvid\_DeathDate\_Date (D, Н) – Дата смерти; * DeathSvid\_DeathDate\_Time (V, Н) – Время смерти; * DeathSvid\_DeathDateStr (D, Н) – Неуточненная дата смерти; * DeathSvid\_isUnknownDeathDate (N, Н) – Признак "Дата смерти неизвестна"; * DeathSvid\_isUnknownDeathTime (N, Н) – Признак "Время смерти неизвестна"; * LpuSection\_id (N, О) – Отделение МО (значение сущности dbo.LpuSection); * MedStaffFact\_id (N, О) – Место работы врача (значение сущности dbo.MedStaffFact); * OrgHead\_id (N, Н) – Руководитель МО (значениесущности dbo.OrgHead); * Person\_mid (N, Н) – Для детей до 1 года: ФИО матери (значение сущности dbo.Person); * Address\_bid (N, Н) – Для детей до 1 года: Место (значение сущности dbo.Address); * ChildTermType\_id (N, Н) – Для детей до 1 года: Доношенность (значение справочника dbo.ChildTermType); * DeathSvid\_Mass (N, Н) – Для детей до 1 года: Масса * DeathSvid\_ChildCount (N, Н) – Для детей до 1 года: Который ребенок (по счету); * DeathSvid\_Month (N, Н) – Для детей до 1 года: Месяц жизни; * DeathSvid\_Day (N, Н) – Для детей до 1 года: День жизни; * Mother\_Age (N, Н) – Для детей до 1 года: Возраст матери; * Mother\_BirthDay (D, Н) – Для детей до 1 года: Дата рождения матери; * DeathEmployment\_id (N, О) – Занятость (значение справочника dbo.DeathEmployment); * DeathEducation\_id (N, О) – Образование (значение справочника dbo.DeathEducation); * DeathPlace\_id (N, Н) – Тип места смерти (значение справочника dbo.DeathPlace); * Address\_did (N, О) – Место смерти (значение справочника dbo.Address); * DeathSvid\_IsNoPla (N, Н) – Признак "Неизввестно место смерти"; * DeathFamilyStatus\_id (N, Н) – Семейное положение (значение справочника dbo.DeathFamilyStatus); * DeathCause\_id (N, О) – Причина смерти (значение справочника dbo.DeathCause); * DeathSvid\_TraumaDate\_Date (D, Н) – Дата начала случая, отравления, травмы; * DeathSvid\_TraumaDate\_Time (V, Н) – Время начала случая, отравления, травмы; * DeathSvid\_TraumaDateStr (D, Н) – Неуточненная дата начала случая, отравления, травмы; * DeathTrauma\_id (N, Н) – Вид травмы (значение справочника dbo.DeathTrauma); * DtpDeathTime\_id (N, Н) – Период наступления смерти от ДТП (значение справочника dbo.DtpDeathTime); * DeathSvid\_TraumaDescr (S, Н) – Место и обстоятельства, при которых произошла травма (отравление); * DeathSetType\_id (N, О) – Кем установлена причина смерти (значение справочника dbo.DeathSetType); * DeathSetCause\_id (N, О) – Основание установления причины смерти (значение справочника dbo.DeathSetCause); * Diag\_iid (N, О) – Непосредственная причина смерти (значение справочника dbo.Diag); * Diag\_tid (N, О) – Патологическое состояние (значение справочника dbo.Diag); * Diag\_mid (N, О) – Первоначальная причина смерти (значение справочника dbo.Diag); * Diag\_eid (N, О) – Внешние причины (значение справочника dbo.Diag); * Diag\_oid (N, Н) – Прочие важные состояния (значение справочника dbo.Diag); * DeathSvid\_Oper (S, Н) – Причины, не связанные с болезнью, а также операции; * Person\_rid (N, Н) – Получатель свидетельства (значение сущности dbo.Person); * DeathSvid\_PolFio (S, Н) – ФИО получателя свидетельства (ручной ввод); * DeathSvid\_RcpDocument (S, Н) - Документ (серия,номер,кем выдан) получателя свидетельства; * DeathSvid\_RcpDate (D, Н) – Дата получения свидетельства. |

* + 1. Создание Свидетельства о смерти

**POST api/DeathSvid**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, О) – Идентификатор человека; * ReceptType\_id (N, О) – Тип свидетельства (значение справочника dbo.ReceptType); * DeathSvid\_Ser (S, О) – Серия свидетельства; * DeathSvid\_Num (N, О) – Номер свидетельства; * DeathSvid\_GiveDate (D, О) – Дата выдачи свидетельства; * DeathSvidType\_id (N, О) – Вид свидетельства (значение справочника dbo.DeathSvidType); * DeathSvid\_OldSer (S, Н) – Серия предыдущего свидетельства; * DeathSvid\_OldNum (N, Н) – Номер предыдущего свидетельства; * DeathSvid\_OldGiveDate (D, Н) – Дата выдачи предыдущего свидетельства; * DeathSvid\_BirthDateStr (D, Н) – Дата рождения; * DeathSvid\_isBirthDate (N, Н) – Признак «Неполная/неизвестная дата рождения»; * DeathSvid\_DeathDate\_Date (D, Н) – Дата смерти; * DeathSvid\_DeathDate\_Time (V, Н) – Время смерти; * DeathSvid\_DeathDateStr (D, Н) – Неуточненная дата смерти; * DeathSvid\_isUnknownDeathDate (N, «.[#99827](https://redmine.swan.perm.ru/issues/99827)О) – Признак «Дата смерти неизвестна». «.[#99827](https://redmine.swan.perm.ru/issues/99827)Возможные значения: 0 и 1, - где 0 – нет, 1 – да; * DeathSvid\_isUnknownDeathTime (N, «.[#99827](https://redmine.swan.perm.ru/issues/99827)О) – Признак «Время смерти неизвестна».».[#99827](https://redmine.swan.perm.ru/issues/99827)Возможные значения: 0 и 1, - где 0 – нет, 1 – да; * LpuSection\_id (N, О) – Отделение МО (значение сущности dbo.LpuSection); * MedStaffFact\_id (N, О) – Место работы врача (значение сущности dbo.MedStaffFact); * OrgHead\_id (N, Н) – Руководитель МО (значениесущности dbo.OrgHead); * Person\_mid (N, Н) – Для детей до 1 года: ФИО матери (значение сущности dbo.Person); * Address\_bid (N, Н) – Для детей до 1 года: Место (значение сущности dbo.Address); * ChildTermType\_id (N, Н) – Для детей до 1 года: Доношенность (значение справочника dbo.ChildTermType); * DeathSvid\_Mass (N, Н) – Для детей до 1 года: Масса * DeathSvid\_ChildCount (N, Н) – Для детей до 1 года: Который ребенок (по счету); * DeathSvid\_Month (N, Н) – Для детей до 1 года: Месяц жизни; * DeathSvid\_Day (N, Н) – Для детей до 1 года: День жизни; * DeathEmployment\_id (N, О) – Занятость (значение справочника dbo.DeathEmployment); * DeathEducation\_id (N, О) – Образование (значение справочника dbo.DeathEducation); * DeathPlace\_id (N, Н) – Тип места смерти (значение справочника dbo.DeathPlace); * Address\_did (N, О) – Место смерти (значение справочника dbo.Address); * DeathSvid\_IsNoPlace (N, «.[#99827](https://redmine.swan.perm.ru/issues/99827)О) – Признак «Неизввестно место смерти».».[#99827](https://redmine.swan.perm.ru/issues/99827)Возможные значения: 0 и 1, - где 0 – нет, 1 – да; * DeathFamilyStatus\_id (N, Н) – Семейное положение (значение справочника dbo.DeathFamilyStatus); * DeathCause\_id (N, О) – Причина смерти (значение справочника dbo.DeathCause); * DeathSvid\_TraumaDate\_Date (D, Н) – Дата начала случая, отравления, травмы; * DeathSvid\_TraumaDate\_Time (V, Н) – Время начала случая, отравления, травмы; * DeathSvid\_TraumaDateStr (D, Н) – Неуточненная дата начала случая, отравления, травмы; * DeathTrauma\_id (N, Н) – Вид травмы (значение справочника dbo.DeathTrauma); * DtpDeathTime\_id (N, Н) – Период наступления смерти от ДТП (значение справочника dbo.DtpDeathTime); * DeathSvid\_TraumaDescr (S, Н) – Место и обстоятельства, при которых произошла травма (отравление); * DeathSetType\_id (N, О) – Кем установлена причина смерти (значение справочника dbo.DeathSetType); * DeathSetCause\_id (N, О) – Основание установления причины смерти (значение справочника dbo.DeathSetCause); * Diag\_iid (N, О) – Непосредственная причина смерти (значение справочника dbo.Diag); * Diag\_tid (N, О) – Патологическое состояние (значение справочника dbo.Diag); * Diag\_mid (N, О) – Первоначальная причина смерти (значение справочника dbo.Diag); * Diag\_eid (N, О) – Внешние причины (значение справочника dbo.Diag); * Diag\_oid (N, Н) – Прочие важные состояния (значение справочника dbo.Diag); * DeathSvid\_Oper (S, Н) – Причины, не связанные с болезнью, а также операции; * Person\_rid (N, Н) – Получатель свидетельства (значение сущности dbo.Person); * DeathSvid\_PolFio (S, Н) – ФИО получателя свидетельства (ручной ввод); * DeathSvid\_RcpDocument (S, Н) – Документ (серия,номер,кем выдан) получателя свидетельства; * DeathSvid\_RcpDate (D, Н) – Дата получения свидетельства. |
| **Ответ** | Производятся проверки:   1. Если по переданному значению Person\_id не найден пациент, то ошибка «Пациент не найден в системе». 2. 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 документа не производится.   1. Если Свидетельство о смерти с аналогичными передаваемыми параметрами для данного человека найдено в системе, создание документа не происходит, ошибка «Данные документа не прошли проверку на дублирование.». 2. Если в передаваемом параметре:  * ReceptType\_id * DeathSvidType\_id * LpuSection\_id * MedStaffFact\_id * OrgHead\_id * Person\_mid * Address\_bid * ChildTermType\_id * DeathEmployment\_id * DeathEducation\_id * DeathPlace\_id * Address\_did * DeathFamilyStatus\_id * DeathCause\_id * DeathTrauma\_id * DtpDeathTime\_id * DeathSetType\_id * DeathSetCause\_id * Diag\_iid * Diag\_tid * Diag\_mid * Diag\_eid * Diag\_oid * Person\_rid   передается значение id, не найденное в системе, ошибка: «Значения для <Наименования полей через запятую> не найдены в справочниках системы.», создание документа не происходит. Аналогичный контроль производится для параметров, по которым ожидается значение 0/1 (да/нет) (DeathSvid\_isUnknownDeathDate, DeathSvid\_isUnknownDeathTime, DeathSvid\_IsNoPlace). Если контроль не пройден, то ошибка «Значение поля <названия полей через запятую> не соответствует ожидаемому», создание документа не производится.   1. Дата смерти DeathSvid\_DeathDate\_Date должна быть больше даты рождения человека. Если проверка не выполняется, отправляется ошибка «Дата смерти должна быть больше даты рождения», создание свидетельства не происходит. 2. Дата получения свидетельства (DeathSvid\_RcpDate) должна быть больше или равна дате выдачи (DeathSvid\_GiveDate). Если проверка не выполняется, ошибка «Дата получения не может быть меньше даты выдачи», создание свидетельства не происходит. 3. Дата/год травмы (DeathSvid\_TraumaDate\_Date) не может быть меньше даты/года рождения человека. Если не выполняется, то ошибка: «Дата/год травмы не может быть меньше даты/года рождения», создание свидетельства не происходит. 4. Период между датой смерти (DeathSvid\_DeathDate\_Date) и датой выдачи свидетельства (DeathSvid\_GiveDate) должен быть меньше 5-ти дней. Если не выполняется, то ошибка «Период между датой смерти и датой выдачи свидетельства должен быть меньше 5-ти дней», создание свидетельства не происходит. 5. Если в одном/всех полях из списка Diag\_iid, Diag\_tid, Diag\_mid, Diag\_eid, передан один из диагнозов «R95.0 Синдром внезапной смерти младенца с упоминанием о вскрытии» и «R95.9 Синдром внезапной смерти младенца без упоминания о вскрытии», то проверяется возраст пациента (человека): он должен быть не больше 11 месяцев 30 дней на дату смерти. Если не выполняется, то ошибка «Для установки диагнозов «R95.0 Синдром внезапной смерти младенца с упоминанием о вскрытии» и «R95.9 Синдром внезапной смерти младенца без упоминания о вскрытии» возраст пациента должен быть не больше 11 месяцев 30 дней на дату смерти», создание свидетельства не происходит. 6. Если в одном/всех полях из списка Diag\_iid, Diag\_tid, Diag\_mid, Diag\_eid, передан диагноз «R54. Старость», то проводится проверка возраста пациента (человека): в год смерти пациента ему(ей) исполнилось или должно было исполнится минимум 81 год. Если не выполняется, то ошибка «Выбор диагноза «R54. Старость» возможен, только если в год смерти пациента ему(ей) исполнилось или должно было исполнится минимум 81 лет», создание свидетельства не происходит. 7. Если дата смерти известна (DeathSvid\_isUnknownDeathDate=нет), то должен быть заполнен один из параметров: DeathSvid\_DeathDate\_Date - Дата смерти либо DeathSvid\_DeathDateStr - Неуточненная дата смерти. Если не выполняется, то ошибка: «Должна быть указана дата смерти либо неуточненная дата смерти, либо указано, что дата смерти неизвестна», создание свидетельства не происходит. 8. Если время смерти известно (DeathSvid\_isUnknownDeathTime=нет), то должен быть заполнен параметр: DeathSvid\_DeathDate\_Time - Время смерти. Если не выполняется, то ошибка: «Не указано время смерти. Необходимо указать точное время смерти, либо указать, что время смерти неизвестно», создание свидетельства не происходит. 9. Для полей:  * Diag\_iid * Diag\_tid * Diag\_mid * Diag\_eid * Diag\_oid   предусмотреть проверку на разрешенные диагнозы – если в поле передан диагноз, не являющийся разрешенным для возраста и пола Человека (смотрим по Person\_id), то ошибка (для каждого поля должна возвращаться отдельная ошибка): «Параметр <название параметра> не прошел проверку на разрешенный диагноз», создание свидетельства не происходит.  Успешный ответ – информация по Свидетельству о смерти:   1. Person\_id (N, О) – Идентификатор человека; 2. DeathSvid\_id (N, О) – Идентификатор Свидетельства о смерти |
| **Пример** |  |

* + 1. Редактирование информации по Свидетельству о смерти

**PUT api/DeathSvid**

|  |  |
| --- | --- |
| **Входящие параметры** | * DeathSvid\_id (N, О) – Идентификатор Свидетельства о смерти; * ReceptType\_id (N, Н) – Тип свидетельства (значение справочника dbo.ReceptType); * DeathSvid\_Ser (S, Н) – Серия свидетельства; * DeathSvid\_Num (N, Н) – Номер свидетельства; * DeathSvid\_GiveDate (D, Н) – Дата выдачи свидетельства; * DeathSvidType\_id (N, Н) – Вид свидетельства (значение справочника dbo.DeathSvidType); * DeathSvid\_OldSer (S, Н) – Серия предыдущего свидетельства; * DeathSvid\_OldNum (N, Н) – Номер предыдущего свидетельства; * DeathSvid\_OldGiveDate (D, Н) – Дата выдачи предыдущего свидетельства; * DeathSvid\_BirthDateStr (D, Н) – Дата рождения; * DeathSvid\_isBirthDate (N, Н) – Признак «Неполная/неизвестная дата рождения»; * DeathSvid\_DeathDate\_Date (D, Н) – Дата смерти; * DeathSvid\_DeathDate\_Time (V, Н) – Время смерти; * DeathSvid\_DeathDateStr (D, Н) – Неуточненная дата смерти; * DeathSvid\_isUnknownDeathDate (N, Н) – Признак «Дата смерти неизвестна»; * DeathSvid\_isUnknownDeathTime (N, Н) – Признак «Время смерти неизвестна»; * LpuSection\_id (N, Н) – Отделение МО (значение сущности dbo.LpuSection); * MedStaffFact\_id (N, Н) – Место работы врача (значение сущности dbo.MedStaffFact); * OrgHead\_id (N, Н) – Руководитель МО (значениесущности dbo.OrgHead); * Person\_mid (N, Н) – Для детей до 1 года: ФИО матери (значение сущности dbo.Person); * Address\_bid (N, Н) – Для детей до 1 года: Место (значение сущности dbo.Address); * ChildTermType\_id (N, Н) – Для детей до 1 года: Доношенность (значение справочника dbo.ChildTermType); * DeathSvid\_Mass (N, Н) – Для детей до 1 года: Масса * DeathSvid\_ChildCount (N, Н) – Для детей до 1 года: Который ребенок (по счету); * DeathSvid\_Month (N, Н) – Для детей до 1 года: Месяц жизни; * DeathSvid\_Day (N, Н) – Для детей до 1 года: День жизни; * DeathEmployment\_id (N, Н) – Занятость (значение справочника dbo.DeathEmployment); * DeathEducation\_id (N, Н) – Образование (значение справочника dbo.DeathEducation); * DeathPlace\_id (N, Н) – Тип места смерти (значение справочника dbo.DeathPlace); * Address\_did (N, Н) – Место смерти (значение справочника dbo.Address); * DeathSvid\_IsNoPlace (N, Н) – Признак «Неизввестно место смерти»; * DeathFamilyStatus\_id (N, Н) – Семейное положение (значение справочника dbo.DeathFamilyStatus); * DeathCause\_id (N, Н) – Причина смерти (значение справочника dbo.DeathCause); * DeathSvid\_TraumaDate\_Date (D, Н) – Дата начала случая, отравления, травмы; * DeathSvid\_TraumaDate\_Time (V, Н) – Время начала случая, отравления, травмы; * DeathSvid\_TraumaDateStr (D, Н) – Неуточненная дата начала случая, отравления, травмы; * DeathTrauma\_id (N, Н) – Вид травмы (значение справочника dbo.DeathTrauma); * DtpDeathTime\_id (N, Н) – Период наступления смерти от ДТП (значение справочника dbo.DtpDeathTime); * DeathSvid\_TraumaDescr (S, Н) – Место и обстоятельства, при которых произошла травма (отравление); * DeathSetType\_id (N, Н) – Кем установлена причина смерти (значение справочника dbo.DeathSetType); * DeathSetCause\_id (N, Н) – Основание установления причины смерти (значение справочника dbo.DeathSetCause); * Diag\_iid (N, Н) – Непосредственная причина смерти (значение справочника dbo.Diag); * Diag\_tid (N, Н) – Патологическое состояние (значение справочника dbo.Diag); * Diag\_mid (N, Н) – Первоначальная причина смерти (значение справочника dbo.Diag); * Diag\_eid (N, Н) – Внешние причины (значение справочника dbo.Diag); * Diag\_oid (N, Н) – Прочие важные состояния (значение справочника dbo.Diag); * DeathSvid\_Oper (S, Н) – Причины, не связанные с болезнью, а также операции; * Person\_rid (N, Н) – Получатель свидетельства (значение сущности dbo.Person); * DeathSvid\_PolFio (S, Н) – ФИО получателя свидетельства (ручной ввод); * DeathSvid\_RcpDocument (S, Н) – Документ (серия,номер,кем выдан) получателя свидетельства; * DeathSvid\_RcpDate (D, Н) – Дата получения свидетельства. |
| **Ответ** | Производятся проверки:   1. Если по переданному значению DeathSvid\_id не найдено свидетельство, то ошибка «Свидетельство о смерти не найдено в системе». 2. 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 документа не производится, обновление данных не происходит.   1. Если в передаваемом параметре:  * ReceptType\_id * DeathSvidType\_id * LpuSection\_id * MedStaffFact\_id * OrgHead\_id * Person\_mid * Address\_bid * ChildTermType\_id * DeathEmployment\_id * DeathEducation\_id * DeathPlace\_id * Address\_did * DeathFamilyStatus\_id * DeathCause\_id * DeathTrauma\_id * DtpDeathTime\_id * DeathSetType\_id * DeathSetCause\_id * Diag\_iid * Diag\_tid * Diag\_mid * Diag\_eid * Diag\_oid * Person\_rid   передается значение id, не найденное в системе, ошибка: «Значения для <Наименования полей через запятую> не найдены в справочниках системы.», обновление данных документа не происходит. Аналогичный контроль производится для параметров, по которым ожидается значение 0/1 (да/нет) (DeathSvid\_isUnknownDeathDate, DeathSvid\_isUnknownDeathTime, DeathSvid\_IsNoPlace). Если контроль не пройден, то ошибка «Значение поля <названия полей через запятую> не соответствует ожидаемому», обновление данных документа не производится.   1. Дата смерти DeathSvid\_DeathDate\_Date (если была передана) должна быть больше даты рождения человека. Если проверка не выполняется, отправляется ошибка «Дата смерти должна быть больше даты рождения», обновления данных не выполняется. 2. Дата получения свидетельства (DeathSvid\_RcpDate) (если была передана) должна быть больше или равна дате выдачи (DeathSvid\_GiveDate). Если проверка не выполняется, ошибка «Дата получения не может быть меньше даты выдачи», обновления данных не выполняется. 3. Дата/год травмы (DeathSvid\_TraumaDate\_Date) (если был передан) не может быть меньше даты/года рождения человека. Если не выполняется, то ошибка: «Дата/год травмы не может быть меньше даты/года рождения», обновления данных не выполняется. 4. Период между датой смерти (DeathSvid\_DeathDate\_Date) и датой выдачи свидетельства (DeathSvid\_GiveDate) должен быть меньше 5-ти дней (если были переданы). Если не выполняется, то ошибка «Период между датой смерти и датой выдачи свидетельства должен быть меньше 5-ти дней», обновления данных не выполняется. 5. Если в одном/всех полях из списка Diag\_iid, Diag\_tid, Diag\_mid, Diag\_eid, передан один из диагнозов «R95.0 Синдром внезапной смерти младенца с упоминанием о вскрытии» и «R95.9 Синдром внезапной смерти младенца без упоминания о вскрытии», то проверяется возраст пациента (человека): он должен быть не больше 11 месяцев 30 дней на дату смерти. Если не выполняется, то ошибка «Для установки диагнозов «R95.0 Синдром внезапной смерти младенца с упоминанием о вскрытии» и «R95.9 Синдром внезапной смерти младенца без упоминания о вскрытии» возраст пациента должен быть не больше 11 месяцев 30 дней на дату смерти», обновления данных не выполняется. 6. Если в одном/всех полях из списка Diag\_iid, Diag\_tid, Diag\_mid, Diag\_eid, передан диагноз «R54. Старость», то проводится проверка возраста пациента (человека): в год смерти пациента ему(ей) исполнилось или должно было исполнится минимум 81 год. Если не выполняется, то ошибка «Выбор диагноза «R54. Старость» возможен, только если в год смерти пациента ему(ей) исполнилось или должно было исполнится минимум 81 лет», обновления данных не выполняется. 7. Если дата смерти известна (DeathSvid\_isUnknownDeathDate=нет), то должен быть заполнен один из параметров: DeathSvid\_DeathDate\_Date - Дата смерти либо DeathSvid\_DeathDateStr - Неуточненная дата смерти. Если не выполняется, то ошибка: «Должна быть указана дата смерти либо неуточненная дата смерти, либо указано, что дата смерти неизвестна», обновления данных не выполняется. 8. Если время смерти известно (DeathSvid\_isUnknownDeathTime=нет), то должен быть заполнен параметр: DeathSvid\_DeathDate\_Time - Время смерти. Если не выполняется, то ошибка: «Не указано время смерти. Необходимо указать точное время смерти, либо указать, что время смерти неизвестно», обновления данных не выполняется. 9. Для полей:  * Diag\_iid * Diag\_tid * Diag\_mid * Diag\_eid * Diag\_oid   предусмотреть проверку на разрешенные диагнозы – если в поле передан диагноз, не являющийся разрешенным для возраста и пола Человека, то ошибка (для каждого поля должна возвращаться отдельная ошибка): «Параметр <название параметра> не прошел проверку на разрешенный диагноз», создание свидетельства не происходит.  Успешный ответ – пустой ответ с кодом ошибки «0» |
| **Пример** |  |

* + 1. Получение списка свидетельств о перинатальной смерти по человеку

**GET api/PntDeathSvidList**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, О) – Идентификатор человека |
| **Ответ** | Успешный ответ – список свидетельств о смерти:   * PntDeathSvid\_id (N, О) – Идентификатор Свидетельства о перинатальной смерти; * PntDeathSvid\_Ser (S, О) – Серия Свидетельства о перинатальной смерти; * PntDeathSvid\_Num (S, О) – Номер Свидетельства о перинатальной смерти; * PntDeathSvid\_GiveDate (D, О) – Дата выдачи Свидетельства о перинатальной смерти |
| **Пример** |  |

* + 1. Получение информации по Свидетельству о перинатальной смерти

**GET api/PntDeathSvid**

|  |  |
| --- | --- |
| **Входящие параметры** | * PntDeathSvid\_id (N, О) – Идентификатор Свидетельства о перинатальной смерти |
| **Ответ** | Успешный ответ – информация по Свидетельству о перинатальной смерти:   * PntDeathSvid\_id (N, О) – Идентификатор Свидетельства; * ReceptType\_id (N, О) – Тип свидетельства (значение справочника dbo.ReceptType); * PntDeathSvid\_Ser (S, О) – Серия свидетельства; * PntDeathSvid\_Num (N, О) – Номер свидетельства; * PntDeathSvid\_GiveDate (D, О) – Дата выдачи свидетельства; * DeathSvidType\_id (N, О) – Вид свидетельства (значение справочника dbo.DeathSvidType); * PntDeathSvid\_OldSer (S, Н) – Серия предыдущего свидетельства; * PntDeathSvid\_OldNum (N, Н) – Номер предыдущего свидетельства; * PntDeathSvid\_OldGiveDate (D, Н) – Дата выдачи предыдущего свидетельства; * PntDeathSvid\_DeathDate (DT, Н) – Дата и время смерти; * PntDeathSvid\_DeathDateStr (D, Н) – Неуточненная дата смерти; * LpuSection\_id (N, О) – Отделение МО (значение сущности dbo.LpuSection); * MedStaffFact\_id (N, О) – Место работы врача (значение сущности dbo.MedStaffFact); * OrgHead\_id (N, Н) – Руководитель МО (значение сущности dbo.OrgHead); * PntDeathSvid\_ChildBirthDT (DT, Н) – Дата и время родов; * PntDeathSvid\_ChildBirthDateStr (D, Н) – Неуточненная дата родов; * PntDeathPeriod\_id (N, Н) – Период смерти (значение справочника dbo.PntDeathPeriod); * PntDeathTime\_id (N, Н) – Время наступления смерти (значение справочника dbo.PntDeathTime); * DeathEmployment\_id (N, О) – Занятость (значение справочника dbo.DeathEmployment); * PntDeathEducation\_id (N, О) – Образование (значение справочника dbo.PntDeathEducation); * PntDeathFamilyStatus\_id (N, Н) – Семейное положение (значение справочника dbo.PntDeathFamilyStatus); * PntDeathSvid\_BirthCount (N, Н) – Которые роды (по счету); * PntDeathSvid\_ChildCount (N, Н) – Который ребенок (по счету); * PntDeathSvid\_ChildFio (S, Н) – ФИО ребенка; * PntDeathPlace\_id (N, О) – Место наступления смерти (значение справочника dbo.PntDeathPlace); * Address\_did (N, О) – Адрес наступления смерти (значение сущности dbo.Address); * PntDeathSvid\_isNoPlace (N, Н) – Признак "Адрес наступления смерти неизвестен"; * Sex\_id (N, О) – Пол ребенка (значение справочника dbo.Sex); * PntDeathGetBirth\_id (N, Н) – Специалист, принимавший роды (значение справочника dbo.PntDeathGetBirth); * PntDeathSvid\_Mass (N, Н) – Масса при рождении, г; * PntDeathSvid\_Height (N, Н) – Рост при рождении, см; * PntDeathSvid\_isMnogoplod (N, Н) – Признак "Многоплодные роды"; * PntDeathSvid\_PlodIndex (N, Н) – Который по счету; * PntDeathSvid\_PlodCount (N, Н) – Всего плодов; * PntDeathCause\_id (N, О) – Причина наступления смерти (значение справочника dbo.PntDeathCause); * PntDeathSvid\_ActNumber (N, Н) – Номер акта о мертворождении; * PntDeathSvid\_ActDt (D, Н) – Дата записи акта о мертворождении; * OrgDep\_id (N, Н) – Орган ЗАГС (значение справочника dbo.OrgDep); * PntDeathSvid\_ZagsFIO (S, Н) – ФИО работника органа ЗАГСа; * PntDeathSvidType\_id (N, Н) – Причина смерти (значение справочника dbo.PntDeathSvidType); * Diag\_iid (N, О) – Основное заболевание ребенка (значение справочника dbo.Diag); * Diag\_tid (N, Н) – Другие заболевания ребенка (значение справочника dbo.Diag); * Diag\_mid (N, Н) – Основное заболевание матери (значение справочника dbo.Diag); * Diag\_eid (N, Н) – Другие заболевания матери (значение справочника dbo.Diag); * Diag\_oid (N, Н) – Другие обстоятельства (значение справочника dbo.Diag); * PntDeathSetType\_id (N, О) – Специалист, установивший причину смерти (значение справочника dbo.PntDeathSetType); * PntDeathSetCause\_id (N, О) – Основание установления причины смерти (значение справочника dbo.PntDeathSetCause); * Person\_rid (N, Н) – Получатель свидетельства (значение сущности dbo.Person); * PntDeathSvid\_PolFio (S, Н) – ФИО получателя свидетельства (ручной ввод); * PntDeathSvid\_RcpDoc (S, Н) – Документ (серия, номер, кем выдан); * DeputyKind\_id (N, Н) – Отношение к ребенку получателя свидетельства (значение справочника dbo.DeputyKind); * DeathSvid\_RcpDate (D, Н) – Дата получения свидетельства; * PntDeathSvid\_RcpDate (N, Н) – Признак "Записано со слов матери". |

* + 1. Создание Свидетельства о перинатальной смерти

**POST api/PntDeathSvid**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, О) – Идентификатор человека; * ReceptType\_id (N, О) – Тип свидетельства (значение справочника dbo.ReceptType); * PntDeathSvid\_Ser (S, О) – Серия свидетельства; * PntDeathSvid\_Num (N, О) – Номер свидетельства; * PntDeathSvid\_GiveDate (D, О) – Дата выдачи свидетельства; * DeathSvidType\_id (N, О) – Вид свидетельства (значение справочника dbo.DeathSvidType); * PntDeathSvid\_OldSer (S, Н) – Серия предыдущего свидетельства; * PntDeathSvid\_OldNum (N, Н) – Номер предыдущего свидетельства; * PntDeathSvid\_OldGiveDate (D, Н) – Дата выдачи предыдущего свидетельства; * PntDeathSvid\_DeathDate (DT, Н) – Дата и время смерти; * PntDeathSvid\_DeathDateStr (D, Н) – Неуточненная дата смерти; * LpuSection\_id (N, О) – Отделение МО (значение сущности dbo.LpuSection); * MedStaffFact\_id (N, О) – Место работы врача (значение сущности dbo.MedStaffFact); * OrgHead\_id (N, Н) – Руководитель МО (значение сущности dbo.OrgHead); * PntDeathSvid\_ChildBirthDT (DT, Н) – Дата и время родов; * PntDeathSvid\_ChildBirthDateStr (D, Н) – Неуточненная дата родов; * PntDeathPeriod\_id (N, Н) – Период смерти (значение справочника dbo.PntDeathPeriod); * PntDeathTime\_id (N, Н) – Время наступления смерти (значение справочника dbo.PntDeathTime); * DeathEmployment\_id (N, О) – Занятость (значение справочника dbo.DeathEmployment); * PntDeathEducation\_id (N, О) – Образование (значение справочника dbo.PntDeathEducation); * PntDeathFamilyStatus\_id (N, Н) – Семейное положение (значение справочника dbo.PntDeathFamilyStatus); * PntDeathSvid\_BirthCount (N, Н) – Которые роды (по счету); * PntDeathSvid\_ChildCount (N, Н) – Который ребенок (по счету); * PntDeathSvid\_ChildFio (S, Н) – ФИО ребенка; * PntDeathPlace\_id (N, О) – Место наступления смерти (значение справочника dbo.PntDeathPlace); * Address\_did (N, О) – Адрес наступления смерти (значение сущности dbo.Address);   PntDeathSvid\_isNoPlace (N, ".О) – Признак "Адрес наступления смерти неизвестен".Возможные значения: 0 и 1, - где 0 – нет, 1 – да;   * Sex\_id (N, О) – Пол ребенка (значение справочника dbo.Sex); * PntDeathGetBirth\_id (N, Н) – Специалист, принимавший роды (значение справочника dbo.PntDeathGetBirth); * PntDeathSvid\_Mass (N, Н) – Масса при рождении, г; * PntDeathSvid\_Height (N, Н) – Рост при рождении, см;   PntDeathSvid\_isMnogoplod (N, ".О) – Признак "Многоплодные роды".".Возможные значения: 0 и 1, - где 0 – нет, 1 – да;   * PntDeathSvid\_PlodIndex (N, Н) – Который по счету; * PntDeathSvid\_PlodCount (N, Н) – Всего плодов; * PntDeathCause\_id (N, О) – Причина наступления смерти (значение справочника dbo.PntDeathCause); * PntDeathSvid\_ActNumber (N, Н) – Номер акта о мертворождении; * PntDeathSvid\_ActDt (D, Н) – Дата записи акта о мертворождении; * OrgDep\_id (N, Н) – Орган ЗАГС (значение справочника dbo.OrgDep); * PntDeathSvid\_ZagsFIO (S, Н) – ФИО работника органа ЗАГСа; * PntDeathSvidType\_id (N, Н) – Причина смерти (значение справочника dbo.PntDeathSvidType); * Diag\_iid (N, О) – Основное заболевание ребенка (значение справочника dbo.Diag); * Diag\_tid (N, Н) – Другие заболевания ребенка (значение справочника dbo.Diag); * Diag\_mid (N, Н) – Основное заболевание матери (значение справочника dbo.Diag); * Diag\_eid (N, Н) – Другие заболевания матери (значение справочника dbo.Diag); * Diag\_oid (N, Н) – Другие обстоятельства (значение справочника dbo.Diag); * PntDeathSetType\_id (N, О) – Специалист, установивший причину смерти (значение справочника dbo.PntDeathSetType); * PntDeathSetCause\_id (N, О) – Основание установления причины смерти (значение справочника dbo.PntDeathSetCause); * Person\_rid (N, Н) – Получатель свидетельства (значение сущности dbo.Person); * PntDeathSvid\_PolFio (S, Н) – ФИО получателя свидетельства (ручной ввод); * PntDeathSvid\_RcpDoc (S, Н) – Документ (серия, номер, кем выдан); * DeputyKind\_id (N, Н) – Отношение к ребенку получателя свидетельства (значение справочника dbo.DeputyKind); * PntDeathSvid\_RcpDate (D, Н) – Дата получения свидетельства;   PntDeathSvid\_IsFromMother (N, ".О) – Признак "Записано со слов матери".".Возможные значения: 0 и 1, - где 0 – нет, 1 – да. |
| **Ответ** | Успешный ответ – информация по Свидетельству о перинатальной смерти:   * Person\_id (N, О) – Идентификатор человека; * PntDeathSvid\_id (N, О) – Идентификатор Свидетельства о перинатальной смерти. |

* + 1. Редактирование информации по Свидетельству о перинатальной смерти

**PUT api/PntDeathSvid**

|  |  |
| --- | --- |
| **Входящие параметры** | * PntDeathSvid\_id (N, О) – Идентификатор Свидетельства о перинатальной смерти; * ReceptType\_id (N, Н) – Тип свидетельства (значение справочника dbo.ReceptType); * PntDeathSvid\_Ser (S, Н) – Серия свидетельства; * PntDeathSvid\_Num (N, Н) – Номер свидетельства; * PntDeathSvid\_GiveDate (D, Н) – Дата выдачи свидетельства; * DeathSvidType\_id (N, Н) – Вид свидетельства (значение справочника dbo.DeathSvidType); * PntDeathSvid\_OldSer (S, Н) – Серия предыдущего свидетельства; * PntDeathSvid\_OldNum (N, Н) – Номер предыдущего свидетельства; * PntDeathSvid\_OldGiveDate (D, Н) – Дата выдачи предыдущего свидетельства; * PntDeathSvid\_DeathDate (DT, Н) – Дата и время смерти; * PntDeathSvid\_DeathDateStr (D, Н) – Неуточненная дата смерти; * LpuSection\_id (N, Н) – Отделение МО (значение сущности dbo.LpuSection); * MedStaffFact\_id (N, Н) – Место работы врача (значение сущности dbo.MedStaffFact); * OrgHead\_id (N, Н) – Руководитель МО (значение сущности dbo.OrgHead); * PntDeathSvid\_ChildBirthDT (DT, Н) – Дата и время родов; * PntDeathSvid\_ChildBirthDateStr (D, Н) – Неуточненная дата родов; * PntDeathPeriod\_id (N, Н) – Период смерти (значение справочника dbo.PntDeathPeriod); * PntDeathTime\_id (N, Н) – Время наступления смерти (значение справочника dbo.PntDeathTime); * DeathEmployment\_id (N, Н) – Занятость (значение справочника dbo.DeathEmployment); * PntDeathEducation\_id (N, Н) – Образование (значение справочника dbo.PntDeathEducation); * PntDeathFamilyStatus\_id (N, Н) – Семейное положение (значение справочника dbo.PntDeathFamilyStatus); * PntDeathSvid\_BirthCount (N, Н) – Которые роды (по счету); * PntDeathSvid\_ChildCount (N, Н) – Который ребенок (по счету); * PntDeathSvid\_ChildFio (S, Н) – ФИО ребенка; * PntDeathPlace\_id (N, Н) – Место наступления смерти (значение справочника dbo.PntDeathPlace); * Address\_did (N, Н) – Адрес наступления смерти (значение сущности dbo.Address); * PntDeathSvid\_isNoPlace (N, Н) – Признак "Адрес наступления смерти неизвестен"; * Sex\_id (N, Н) – Пол ребенка (значение справочника dbo.Sex); * PntDeathGetBirth\_id (N, Н) – Специалист, принимавший роды (значение справочника dbo.PntDeathGetBirth); * PntDeathSvid\_Mass (N, Н) – Масса при рождении, г; * PntDeathSvid\_Height (N, Н) – Рост при рождении, см; * PntDeathSvid\_isMnogoplod (N, Н) – Признак "Многоплодные роды"; * PntDeathSvid\_PlodIndex (N, Н) – Который по счету; * PntDeathSvid\_PlodCount (N, Н) – Всего плодов; * PntDeathCause\_id (N, Н) – Причина наступления смерти (значение справочника dbo.PntDeathCause); * PntDeathSvid\_ActNumber (N, Н) – Номер акта о мертворождении; * PntDeathSvid\_ActDt (D, Н) – Дата записи акта о мертворождении; * OrgDep\_id (N, Н) – Орган ЗАГС (значение справочника dbo.OrgDep); * PntDeathSvid\_ZagsFIO (S, Н) – ФИО работника органа ЗАГСа; * PntDeathSvidType\_id (N, Н) – Причина смерти (значение справочника dbo.PntDeathSvidType); * Diag\_iid (N, Н) – Основное заболевание ребенка (значение справочника dbo.Diag); * Diag\_tid (N, Н) – Другие заболевания ребенка (значение справочника dbo.Diag); * Diag\_mid (N, Н) – Основное заболевание матери (значение справочника dbo.Diag); * Diag\_eid (N, Н) – Другие заболевания матери (значение справочника dbo.Diag); * Diag\_oid (N, Н) – Другие обстоятельства (значение справочника dbo.Diag); * PntDeathSetType\_id (N, Н) – Специалист, установивший причину смерти (значение справочника dbo.PntDeathSetType); * PntDeathSetCause\_id (N, Н) – Основание установления причины смерти (значение справочника dbo.PntDeathSetCause); * Person\_rid (N, Н) – Получатель свидетельства (значение сущности dbo.Person); * PntDeathSvid\_PolFio (S, Н) – ФИО получателя свидетельства (ручной ввод); * PntDeathSvid\_RcpDoc (S, Н) – Документ (серия, номер, кем выдан); * DeputyKind\_id (N, Н) – Отношение к ребенку получателя свидетельства (значение справочника dbo.DeputyKind); * PntDeathSvid\_RcpDate (D, Н) – Дата получения свидетельства; * PntDeathSvid\_IsFromMother (N, Н) – Признак "Записано со слов матери". |
| **Ответ** | * Успешный ответ – пустой ответ с кодом ошибки «0» |

* 1. Описание методов работы с прочими документами (шаблоны, эпикризы)
  2. 1. Получение списка документов случая

**GET api/XmlDocumentList**

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_id (N, О) – идентификатор случая |
| **Ответ** | Успешный ответ – список документов случая:   * Evn\_id (N, О) – идентификатор случая; * EvnXml\_id (N, О) – идентификатор документа; * XmlType\_id (N, О) – Тип документа (значение справочника dbo.XmlType) |

* + 1. Получение содержимого документа

**GET api/XmlDocument**

|  |  |
| --- | --- |
| **Запрос** | * Evn\_id (N, Н) – идентификатор случая; * EvnXml\_id (N, Н) – идентификатор документа; * XmlType\_id (N, Н) – Тип документа (значение справочника dbo.XmlType) |
| **Ответ** | Если не передан ни один параметр, то ошибка.  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список документов с информацией:   * Evn\_id (N, О) – идентификатор случая; * EvnXml\_id (N, О) – идентификатор документа; * XmlType\_id (N, О) – Тип документа (значение справочника dbo.XmlType); * EvnXml\_Data (Xml, О) – Текст документа в Xml; * XmlTemplateHtml\_HtmlTemplate (S, Н) – Текст документа в Html * XmlTemplate\_id (N, О) – Идентификатор шаблона документа |

* + 1. Создание документа

**POST api/XmlDocument**

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_id (N, О) – идентификатор случая; * XmlType\_id (N, О) – Тип документа (значение справочника dbo.XmlType); * EvnXml\_Data (Xml, О) – Текст документа в Xml; * XmlTemplateHtml\_HtmlTemplate (S, Н) – Текст документа в Html * EvnXml\_Data64 (Base64, Н) – Текст документа в Xml закодированный в base64. Запись в поле EvnXml\_Data после распаковки; * XmlTemplateHtml\_HtmlTemplate64 (Base64, Н) – Текст документа в Html закодированный в base64. Запись в поле XmlTemplateHtml\_HtmlTemplate после распаковки. * XmlTemplate\_id (N, О) – Идентификатор шаблона документа |
| **Ответ** | * Evn\_id (N, О) – идентификатор случая; * EvnXml\_id (N, О) – идентификатор документа |

* + 1. Редактирование документа

**PUT api/XmlDocument**

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_id (N, О) – идентификатор случая; * EvnXml\_id (N, О) – идентификатор документа; * XmlType\_id (N, Н) – Тип документа (значение справочника dbo.XmlType); * EvnXml\_Data (Xml, Н) – Текст документа в Xml; * XmlTemplateHtml\_HtmlTemplate (S, Н) – Текст документа в Html * EvnXml\_Data64 (Base64, Н) – Текст документа в Xml закодированный в base64. Запись в поле EvnXml\_Data после распаковки; * XmlTemplateHtml\_HtmlTemplate64 (Base64, Н) – Текст документа в Html закодированный в base64. Запись в поле XmlTemplateHtml\_HtmlTemplate после распаковки. |
| **Ответ** | Если документ найден и отредактирован, то формируется успешный ответ – пустой ответ с кодом ошибки «0» |

* 1. Описание методов работы с назначениями медикаментов
  2. 1. Получение списка назначений медикаментов в рамках случая

**GET api/EvnDrugList**

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_pid (N, О) – Идентификатор случая-родителя (движение в КВС, посещения в ТАП) |
| **Ответ** | * Evn\_id (N, О) – Идентификатор случая назначения медикамента; * Evn\_pid (N, О) – Идентификатор случая-родителя (движение в КВС, посещения в ТАП); * Evn\_setDt (DT, О) – Дата и время назначения; * LpuSection\_id (N, О) – Отделение МОЛ (значение сущности dbo.LpuSection); * Mol\_id (N, О) – Материально-ответственное лицо (значение сущности dbo.Mol); * Drug\_id (N, О) – Медикамент (значение справочника rls.Drug); * EvnDrug\_Kolvo (N, О) – Кол-во (ед. уч.); * EvnDrug\_KolvoEd (N, О) – Кол-во (ед. спис.) |

* + 1. Получение информации по назначению медикамента

**GET api/EvnDrug**

|  |  |
| --- | --- |
| **Запрос** | * Evn\_id (N, Н) – Идентификатор случая назначения медикамента; * Evn\_pid (N, Н) – Идентификатор случая-родителя (движение в КВС, посещения в ТАП); * Evn\_setDt (DT, Н) – Дата и время назначения; * Drug\_id (N, Н) – Медикамент (значение справочника rls.Drug) |
| **Ответ** | Если не передан ни один параметр, то ошибка.  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список назначений медикаментов с информацией:   * Evn\_id (N, О) – Идентификатор случая назначения медикамента; * Evn\_pid (N, О) – Идентификатор случая-родителя (движение в КВС, посещения в ТАП); * Evn\_setDt (DT, О) – Дата и время назначения; * LpuSection\_id (N, О) – Отделение МОЛ (значение сущности dbo.LpuSection); * Mol\_id (N, О) – Материально-ответственное лицо (значение сущности dbo.Mol); * Drug\_id (N, О) – Медикамент (значение справочника rls.Drug); * EvnDrug\_Kolvo (N, О) – Кол-во (ед. уч.); * EvnDrug\_KolvoEd (N, О) – Кол-во (ед. спис.) |

* + 1. Создание назначения медикамента

**POST api/EvnDrug**

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_pid (N, О) – идентификатор случая, в рамках которого назначается медикамент (движение КВС, посещение ТАП); * Evn\_setDt (DT, О) – Дата и время назначения; * LpuSection\_id (N, О) – Отделение МОЛ (значение сущности dbo.LpuSection); * Mol\_id (N, О) – Материально-ответственное лицо (значение сущности dbo.Mol); * Drug\_id (N, О) – Медикамент (значение справочника rls.Drug); * EvnDrug\_Kolvo (N, О) – Кол-во (ед. уч.); * EvnDrug\_KolvoEd (N, О) – Кол-во (ед. спис.) |
| **Ответ** | * Evn\_id (N, О) – Идентификатор случая назначения медикамента |

* + 1. Редактирование назначения медикамента

**PUT api/EvnDrug**

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_id (N, О) – Идентификатор случая назначения медикамента; * Evn\_setDt (DT, Н) – Дата и время назначения; * LpuSection\_id (N, Н) – Отделение МОЛ (значение сущности dbo.LpuSection); * Mol\_id (N, Н) – Материально-ответственное лицо (значение сущности dbo.Mol); * Drug\_id (N, Н) – Медикамент (значение справочника rls.Drug); * EvnDrug\_Kolvo (N, Н) – Кол-во (ед. уч.); * EvnDrug\_KolvoEd (N, Н) – Кол-во (ед. спис.) |
| **Ответ** | Если документ найден и отредактирован, то формируется успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Описание методов работы с назначениями в КВС/ТАП
       1. Назначение режима
          1. Получить назначение по идентификатору назначения

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnPrescrRegime\_id (N,O) – идентификатор c типом назначения «Режим» * TimeTableGraf\_id (N, У) ссылка на бирку поликлиники * TimeTablePar\_id (N,У) ссылка на бирку параклиники * TimeTableStac\_id (N,У) ссылка на бирку стационара * PrescriptionRegimeType\_id (N,O) -Тип режима * EvnPrescr\_setDate (N,O) - Дата начала * EvnPrescr\_dayNum (N,O) – Продолжать * EvnPrescr\_IsExec (N,O) признак выполнения (Да/Нет). |
| **Ответ** | * TimeTableGraf\_id (N, У) ссылка на бирку поликлиники * TimeTablePar\_id (N,У) ссылка на бирку параклиники * TimeTableStac\_id (N,У) ссылка на бирку стационара * PrescriptionRegimeType\_id (N,O) -Тип режима * EvnPrescr\_setDate (N,O) - Дата начала * EvnPrescr\_dayNum (N,O) – Продолжать   EvnPrescr\_IsExec (N,O) признак выполнения (Да/Нет |

* + - * 1. Создать назначение для ТАП/КВС

|  |  |
| --- | --- |
| **Входящие параметры** | * TimeTableGraf\_id (N, У) ссылка на бирку поликлиники * TimeTablePar\_id (N,У) ссылка на бирку параклиники * TimeTableStac\_id (N,У) ссылка на бирку стационара * PrescriptionRegimeType\_id (N,O) -Тип режима * EvnPrescr\_setDate (N,O) - Дата начала * EvnPrescr\_dayNum (N,O) – Продолжать   EvnPrescr\_IsExec (N,O) признак выполнения (Да/Нет) Возможные значения: 0 и 1, - где 0 – нет, 1 – да. |
| **Ответ** | EvnPrescrRegime\_id (N,O) – идентификатор c типом назначения «Режим» |

* 1. Описание методов работы с КВС
  2. 1. Получить КВС

**GET api/EvnPS**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnPS\_id (N,Н) Идентификатор КВС; * Evn\_id (N, Н) – Идентификатор случая; * Person\_id (N,Н) – ссылка на человека в Person; * EvnPS\_NumCard (T,Н) – номер карты; * Evn\_setDT (D,Н) – Дата поступления. |
| **Ответ** | Если не передан ни один параметр, то ошибка.  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список КВС с информацией:   * EvnPS\_id (N,Н) Идентификатор КВС; * Evn\_id (N, Н) – Идентификатор случая; * Person\_id (N,O) ссылка на человека в Person * Lpu\_id (N,O) – Идентификатор МО * EvnPS\_IsCont (N,O) – Продолжение случая * EvnPS\_NumCard (T,O) – номер карты * PayType\_id (N,O) – Тип оплаты * Evn\_setDT (D,O) – Дата и время поступления * EvnPS\_IsWithoutDirection (N,O) – С электронным направлением * EvnPS\_IsImperHosp (N,O) – Несвоевременность госпитализации * EvnPS\_IsShortVolume (N,O) – Недостаточный обьем оперативной помощи * EvnPS\_IsWrongCure (N,O) – Неправильная тактика лечения * EvnPS\_IsDiagMismatch (N,O) – Несовпадение диагноза * PrehospType\_id (N,O) – идентификатор типа предварительной госпитализации * PrehospDirect\_id (N,У) – кем направлен в предварительной госпитализации. Если тип предварительной госпитализации «Плановый», то атрибут обязательный * LpuSection\_did (N,У) – идентификатор направившего отделения; * Lpu\_did (N,У) – идентификатор направившей МО; * Org\_did (N,У) – идентификатор направившей организации; * OrgMilitary\_did(N,У) – идентификатор направившего военкомата; * PrehospArrive\_id (N,Н) – кем доставлен при предварительной госпитализации * CmpCallCard\_id (N,H) – идентификатор Карты вызова * EvnPS\_CodeConv (T, H) – Кем доставлен (код) * EvnPS\_NumConv (T,H) – Кем доставлен (номер наряда) * EvnPS\_IsPLAmbulance (N,H) – признак «Талон передан на ССМП» (Да/Нет) * EvnPS\_IsWaif (N,O) – признак Беспризорный (Да/Нет) Если несовершеннолетний * PrehospWaifArrive\_id (N,Y) – Кем доставлен, если беспризорный * PrehospWaifReason\_id (N,Y) – Причина помещения в ЛПУ, если беспризорный * PrehospToxic\_id(N,H) – Состояние опьянения (вид отравления) * EvnPS\_HospCount (N,H) – количество госпитализаций * Okei\_id (N,O) – Ссылка на общероссийский классификатор единиц измерения (единицы измерения для времени с начала заболевания) * EvnPS\_TimeDesease (N,H) – Время с начала заболевания * EvnPS\_IsNeglectedCase(N,H) – Случай запущен * PrehospTrauma\_id (N, H) –Вид травмы (внешнего воздействия) травма при предварительной госпитализации * EvnPS\_IsUnlaw (N,H) – противоправная * EvnPS\_IsUnport (N,H) – нетранспортабельность * LpuSection\_pid (N,У) – Приемное отделение * MedPersonal\_pid (N,У) – Врач приемного отделения * Diag\_pid (N,У) – Основной диагноз приемного отделения * DiagSetPhase\_pid(N,H) – Стадия/Фаза заболевания для диагнозов приемного (Diag\_pid) * Diag\_id (N,У) – Основной диагноз * Diag\_eid (N,H) – Внешняя причина; * Diag\_aid (N,H) – Паталогоанатомический диагноз; * Diag\_did (N,H) – Основной диагноз направившего учреждения; * Diag\_pid (N,H) – Основной диагноз приемного отделения: * EvnPS\_PhaseDescr\_pid(N,H) – Описание фазы для диагнозов приемного * DeseaseType\_id (N, У) – Характер (справочник DeseaseType), обязательный параметр, если основной диагноз приемного отделения (Diag\_pid) указан отличный от Z00-Z99, дата исхода (EvnPS\_OutcomeDT) больше или равна 01.11.2018 и указана причина отказа от госпитализации * EvnPS\_IsPrehospAcceptRefuse (N, У) – Отказ в подтверждении госпитализации (PrehospWaifRefuseCause\_id) * PrehospWaifRefuseCause\_id (N, Н) – Причина отказа от госпитализации (справочник PrehospWaifRefuseCause) * EvnPS\_OutcomeDT (D, У) – Дата исхода в приемном отделении * LpuSection\_eid (N,У) – Идентификатор отделения для госпитализации; * LpuSectionProfile\_id (N,У) – Идентификатор профиля отделения; * UslugaComplex\_id(N.У) – Код посещения; * EvnDirection\_id (N, H) – Идентификатор направления на госпитализацию. |
| **Пример** | Запрос  Ответ |

* + 1. Получение списка движений для КВС

**GET api/EvnSectionList**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnPS\_id (N,O) Идентификатор КВС |
| **Ответ** | Успешный ответ - список движений для КВС с указанием следующих параметров   * EvnSection\_id (N,O) идентификатор движения * EvnPS\_NumCard (T,O) - номер карты * Lpu\_id (N,O) - Идентификатор МО |

* + 1. Получение списка диагнозов (основного, сопутствующих или осложнений) по движению

**GET api/EvnDiagPSByPar**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnSection\_id (N,O) идентификатор движения * DiagSetClass\_id (N,Н) вид диагноза. |
| **Ответ** | Успешный ответ - список сопутствующих диагнозов или осложнений для движения с указанием следующих параметров   * EvnDiagPS\_id (N,O) – идентификатор диагноза * EvnDiagPS\_setDate (D,O) дата установки * EvnDiagPS\_setTime(V,H) – время установки * DiagSetClass\_id (N,O) вид диагноза * DiagSetType\_id (N,O) Тип диагноза * Diag\_id (N,O) Диагноз * DiagSetPhase\_id (N,H) Стадия/фаза * EvnDiagPS\_PhaseDescr (T,H) Расшифровка |

* + 1. Получить движение

**GET api/EvnSection**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnSection\_id (N,Н) идентификатор движения; * Evn\_id (N, Н) – Идентификатор случая движения; * EvnPS\_id (N,Н) Идентификатор КВС * EvnSection\_setDate (D,Н) – Дата поступления; * Date\_DT (D, Н) – Дата (для фильтрации движений по дате, входящей в период движения); * EvnSection\_IsPriem (N, Н) – признак «Приемное отделение». |
| **Ответ** | Если не передан ни один параметр, то ошибка.  Если не передан ни один из параметров Evn\_id или EvnPS\_id, то ошибка.  Если не найдено ни одной записи, то ответ «0».  Если задан входной параметр Date\_DT, то фильтруются записи (движения), по которым Date\_DT входит в период между EvnSection\_setDate и EvnSection\_disDate.  Если хотя бы одна запись найдена, то успешный ответ – список движений с информацией:   * EvnSection\_id (N,Н) идентификатор движения; * Evn\_id (N, Н) – Идентификатор случая движения; * EvnPS\_id (N,O) Идентификатор КВС * Evn\_setDT – (D,O) Дата поступления * Evn\_disDT – (D,O) Дата выписки из отделения; * PayType\_id (N,O) – Тип оплаты * TariffClass\_id (N,H) – Вид (класс) тарифа * LpuSection\_id (N,O) Отделение ЛПУ * LpuSectionWard\_id (N,H) – палатная структура * MedStaffFact\_id (N,O) – врач * Diag\_id (N,O) – основной диагноз * DeseaseType\_id (N, У) – Характер (справочник DeseaseType), обязательный параметр, если основной диагноз (Diag\_id) отличный от Z00-Z99 и дата поступления или дата выписки (Evn\_setDT, Evn\_disDT) больше или равна 01.11.2018. * DiagSetPhase\_id (N,Н) – стадия/фаза (для основного диагноза в движении) * EvnSection\_PhaseDescr (T,H) – расшифровка (описание фазы) * Mes\_id (N,Н) – идентификатор МЭС * Mes\_sid(N,H) – КСГ * LpuSectionProfile\_id (N,O) – Профиль * LeaveType\_id (N, У) – Исход госпитализации * EvnLeave\_UKL(N,У) – Уровень качества лечения * ResultDesease\_id(N,У) – Исход заболевания * LeaveCause\_id (N,У) – Причина (выписки, перевода) * CureResult\_id (N,У) ***Регионы: Астрахань, Карелия, Крым, Пермь*** – Идентификатор Итога лечения (поле «Итог лечения» в форме «Установка случая движения»). Обязателен при наличии Исхода госпитализации. * EvnLeave\_IsAmbul(N,У) – направлен на амбулаторное лечение * Org\_oid(N,У) – Идентификатор МО для перевода * MedStaffFact\_did(N,У) – врач установивший смерть * EvnDie\_IsAnatom(N,У) – необходимость экспертизы (если смерть) * LpuUnitType\_oid(N,У) – тип стационара при переводе (если «Исход госпитализации = Переведен в стационар) * LpuSection\_oid (N, У) – Отделение переводе (если «Исход госпитализации = Переведен в стационар или Переведен на другой профильк коек) * LpuSectionBedProfile\_oid (N,O) – профиль коек (если результат госпитализации = переведен на другой профиль коек) * EvnSection\_IsPriem(N,О) – Признак приемного отделения. * LpuSectionBedProfile\_id (N,H) – Идентификатор профиля койки. * LpuSectionBedProfileLink\_fedid (N,У) – Идентификатор связки между региональным профилем коек и федеральным профилем коек (поле «Профиль коек» в форме «Установка случая движения»). |
| **Пример** | Запрос  Ответ |

* + 1. Получить установленный диагноз в стационаре (сопутсвтующий, осложнение) по идентификатору диагноза

**GET api/EvnDiagPS**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnDiagPS\_id (N,O) - идентификатор диагноза |
| **Ответ** | Успешный ответ -   * EvnSection\_id (N,O) идентификатор движения * EvnDiagPS\_setDate (D,O) дата установки * EvnDiagPS\_setTime(V,H) – время установки * DiagSetClass\_id (N,O) вид диагноза * DiagSetType\_id (N,O) Тип диагноза * Diag\_id (N,O) Диагноз * DiagSetPhase\_id (N,H) Стадия/фаза * EvnDiagPS\_PhaseDescr (T,H) Расшифровка |

* + 1. Создать КВС (c приемным)

**POST api/EvnPS**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N,O) ссылка на человека в Person * Lpu\_id (N,O) – Идентификатор МО   EvnPS\_IsCont (N,O) – Продолжение случая.  Возможные значения: 0 и 1, - где 0 – нет, 1 – да.   * EvnPS\_NumCard (T,O) – номер карты * PayType\_id (N,O) – Тип оплаты * Evn\_setDT (D,O) – Дата и время поступления   EvnPS\_IsWithoutDirection (N,O) – С электронным направлением. Возможные значения: 0 и 1, - где 0 – нет, 1 – да.   * EvnDirection\_Num (T, Н) – Номер направления; * EvnDirection\_setDT (D, Н) – Дата направления. * EvnPS\_IsImperHosp (N,O) – Несвоевременность госпитализации. Возможные значения: 0 и 1, - где 0 – нет, 1 – да. * EvnPS\_IsShortVolume (N,O) – Недостаточный обьем оперативной Возможные значения: 0 и 1, - где 0 – нет, 1 – да.   EvnPS\_IsWrongCure (N,O) – Неправильная тактика лечения.  Возможные значения: 0 и 1, - где 0 – нет, 1 – да.  EvnPS\_IsDiagMismatch (N,O) – Несовпадение диагноза.  Возможные значения: 0 и 1, - где 0 – нет, 1 – да.   * PrehospType\_id (N,O) – идентификатор типа предварительной госпитализации * PrehospDirect\_id (N,У) – кем направлен в предварительной госпитализации. Если тип предварительной госпитализации «Плановый», то атрибут обязательный * LpuSection\_did (N,У) –идентификатор направившего отделения; * Lpu\_did (N,У) –идентификатор направившей МО; * Org\_did (N,У) – идентификатор направившей организации; * OrgMilitary\_did(N,У) – идентификатор направившего военкомата; * PrehospArrive\_id (N,Н) – кем доставлен при предварительной госпитализации * CmpCallCard\_id (N,H) – идентификатор Карты вызова * EvnPS\_CodeConv (T, H) –Кем доставлен (код) * EvnPS\_NumConv (T,H) –Кем доставлен (номер наряда) * EvnPS\_IsPLAmbulance (N,У) – признак «Талон передан на ССМП» (Да/Нет)   Возможные значения: 0 и 1, - где 0 – нет, 1 – да.  EvnPS\_IsWaif (N,O) – признак Беспризорный (Да/Нет) Если несовершеннолетний.  Возможные значения: 0 и 1, - где 0 – нет, 1 – да.   * PrehospWaifArrive\_id (N,Y) – Кем доставлен, если беспризорный * PrehospWaifReason\_id (N,Y) – Причина помещения в ЛПУ, если беспризорный * PrehospToxic\_id(N,H) –Состояние опьянения (вид отравления) * EvnPS\_HospCount (N,H) – количество госпитализаций * Okei\_id (N,O) – Ссылка на общероссийский классификатор единиц измерения (единицы измерения для времени с начала заболевания) * EvnPS\_TimeDesease (N,H) –Время с начала заболевания * EvnPS\_IsNeglectedCase(N,H) –Случай запущен * PrehospTrauma\_id (N, H) –Вид травмы (внешнего воздействия) травма при предварительной госпитализации * EvnPS\_IsUnlaw (N, О) –противоправная. Возможные значения: 0 и 1, - где 0 – нет, 1 – да. * EvnPS\_IsUnport (N, О) –нетранспортабельность. <https://redmine.swan.perm.ru/issues/99827>Возможные значения: 0 и 1, - где 0 – нет, 1 – да. * LpuSection\_pid (N,У) – Приемное отделение * MedPersonal\_pid (N,У) – Врач приемного отделения * Diag\_pid (N,У) – Основной диагноз приемного отделения * DiagSetPhase\_pid(N,H) –Стадия/Фаза заболевания для диагнозов приемного (Diag\_pid) * EvnPS\_PhaseDescr\_pid(N,H) – Описание фазы для диагнозов приемного * DeseaseType\_id (N, У) – Характер (справочник DeseaseType), обязательный параметр, если основной диагноз приемного отделения (Diag\_pid) указан отличный от Z00-Z99, дата исхода (EvnPS\_OutcomeDT) больше или равна 01.11.2018 и указана причина отказа от госпитализации * PrehospWaifRefuseCause\_id (N, Н) - Причина отказа от госпитализации (справочник PrehospWaifRefuseCause) * EvnPS\_IsPrehospAcceptRefuse (N, О) – Отказ в подтверждении госпитализации. Возможные значения: 0 и 1, - где 0 – нет, 1 – да. * EvnPS\_OutcomeDT (D, У) – Дата исхода в приемном отделении * LpuSection\_eid (N,У) – Идентификатор отделения для госпитализации; * LpuSectionProfile\_id (N,У) – Идентификатор профиля отделения; * UslugaComplex\_id (N.У) – Код посещения; |
| **Ответ** | Успешный ответ – 0   * EvnPS\_id (N,O) Идентификатор КВС * EvnSection\_id (N,Н) идентификатор движения (для приемного) |
| **Пример** | Запрос  Ответ |

* + 1. Создать движение в отделение

**POST api/EvnSection**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnPS\_id (N,O) Идентификатор КВС * Evn\_setDT (D,O) Дата и время поступления * Evn\_disDT (D,У) –Дата выписки; * PayType\_id (N,O) – Тип оплаты * TariffClass\_id (N,H) – Вид (класс) тарифа * LpuSection\_id (N,O) Отделение ЛПУ * LpuSectionWard\_id (N,H) – палатная структура * MedStaffFact\_id (N,O) – врач * Diag\_id (N,O) – основной диагноз * DiagSetPhase\_id (N,Н) – стадия/фаза (для основного диагноза в движении) * EvnSection\_PhaseDescr (T,H) – расшифровка (описание фазы) * Mes\_id (N,Н) – идентификатор МЭС * Mes\_sid(N,H) – КСГ * LpuSectionProfile\_id (N,O) –Профиль * LeaveType\_id (N, У) –Исход госпитализации * CureResult\_id (N,У) ***Регионы: Астрахань, Карелия, Крым, Пермь***– Идентификатор итога лечения. Обязателен при наличии Исхода госпитализации. * EvnLeave\_UKL(N,У) – Уровень качества лечения * ResultDesease\_id(N,У) – Исход заболевания * LeaveCause\_id (N,У) – Причина (выписки, перевода) * EvnLeave\_IsAmbul(N,У) – направлен на амбулаторное лечение * Org\_oid(N,У) – Идентификатор МО для перевода * MedStaffFact\_did(N,У) – врач установивший смерть * EvnDie\_IsAnatom(N,У) –необходимость экспертизы (если смерть) * LpuUnitType\_oid(N,У) – тип стационара при переводе (если «Исход госпитализации = Переведен в стационар) * LpuSection\_oid (N, У) – Отделение переводе (если «Исход госпитализации = Переведен в стационар или Переведен на другой профильк коек) * LpuSectionBedProfile\_oid (N,O) – профиль коек (если результат госпитализации = переведен на другой профиль коек) * LpuSectionBedProfileLink\_fedid (N,У) – Идентификатор связки между региональным профилем коек и федеральным профилем коек. * EvnSection\_IsPriem(N,О) –Признак приемного отделения (да/нет). Возможные значения: 0 и 1, - где 0 – нет, 1 – да; * DeseaseType\_id (N, У) – Характер (справочник DeseaseType), обязательный параметр, если основной диагноз (Diag\_id) отличный от Z00-Z99 и дата поступления или дата выписки (Evn\_setDT, Evn\_disDT) больше или равна 01.11.2018. * EvnPS\_IsPrehospAcceptRefuse (N, У) – Отказ в подтверждении госпитализации (PrehospWaifRefuseCause\_id) |
| **Ответ** | Успешный ответ – 0   * EvnSection\_id (N,O) идентификатор движения |
| **Пример** | Запрос  Ответ |

* + 1. Установка диагноза пациенту в стационаре: Добавление (Сопутствующие диагнозы) или осложнения

**POST api/EvnDiagPS**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnSection\_id (N,O) идентификатор движения * EvnDiagPS\_setDate (D,O) дата установки * EvnDiagPS\_setTime(V,H) – время установки * DiagSetClass\_id (N,O) вид диагноза * DiagSetType\_id (N,O) Тип диагноза * Diag\_id (N,O) Диагноз * DiagSetPhase\_id (N,H) Стадия/фаза * EvnDiagPS\_PhaseDescr (T,H) Расшифровка |
| **Ответ** | Успешный ответ - 0   * EvnDiagPS\_id (N,O) – идентификатор диагноза |

* + 1. Редактировать КВС

**PUT api/EvnPS**

|  |  |
| --- | --- |
|  | * EvnPS\_id (N,O) Идентификатор КВС * Person\_id (N,У) ссылка на человека в Person * EvnPS\_IsCont (N,У) – Продолжение случая * EvnPS\_NumCard (T,У) – номер карты * PayType\_id (N,У) – Тип оплаты * Evn\_setDT (D,У) – Дата и время поступления * EvnPS\_IsWithoutDirection (N,У) – С электронным направлением * EvnDirection\_Num (T, Н) – Номер направления; * EvnDirection\_setDT (D, Н) – Дата направления. * EvnPS\_IsImperHosp (N,У) – Несвоевременность госпитализации * EvnPS\_IsShortVolume (N,У) – Недостаточный обьем оперативной помощи * EvnPS\_IsWrongCure (N,У) – Неправильная тактика лечения * EvnPS\_IsDiagMismatch (N,У) – Несовпадение диагноза * PrehospType\_id (N,У) – идентификатор типа предварительной госпитализации * PrehospDirect\_id (N,У) – кем направлен в предварительной госпитализации. Если тип предварительной госпитализации «Плановый», то атрибут обязательный * PrehospArrive\_id (N,У) – кем доставлен при предварительной госпитализации * CmpCallCard\_id (N,У) – идентификатор Карты вызова * EvnPS\_CodeConv (T, У) –Кем доставлен (код) * EvnPS\_NumConv (T,У) –Кем доставлен (номер наряда) * EvnPS\_IsPLAmbulance (N,У) – признак «Талон передан на ССМП» (Да/Нет) * EvnPS\_IsWaif (N,У) – признак Беспризорный (Да/Нет) Если несовершеннолетний * PrehospWaifArrive\_id (N,У) – Кем доставлен, если беспризорный * PrehospWaifReason\_id (N,У) – Причина помещения в ЛПУ, если беспризорный * PrehospToxic\_id(N,У) –Состояние опьянения (вид отравления) * EvnPS\_HospCount (N,У) – количество госпитализаций * Okei\_id (N,У) – Ссылка на общероссийский классификатор единиц измерения (единицы измерения для времени с начала заболевания) * EvnPS\_TimeDesease (N,У) –Время с начала заболевания * EvnPS\_IsNeglectedCase(N,У) –Случай запущен * PrehospTrauma\_id (N, У) –Вид травмы (внешнего воздействия) травма при предварительной госпитализации * EvnPS\_IsUnlaw (N,У) –противоправная * EvnPS\_IsUnport (N,У) –нетранспортабельность * LpuSection\_pid (N,У) – Приемное отделение * MedPersonal\_pid (N,У) – Врач приемного отделения * Diag\_pid (N,У) – Основной диагноз приемного отделения * DiagSetPhase\_pid(N,У) –Стадия/Фаза заболевания для диагнозов приемного (Diag\_pid) * EvnPS\_PhaseDescr\_pid(N,У) –Описание фазы для диагнозов приемного * DeseaseType\_id (N, У) – Характер (справочник DeseaseType), обязательный параметр, если основной диагноз приемного отделения (Diag\_pid) указан отличный от Z00-Z99, дата исхода (EvnPS\_OutcomeDT) больше или равна 01.11.2018 и указана причина отказа от госпитализации * EvnPS\_IsPrehospAcceptRefuse (N, У) – Отказ в подтверждении госпитализации (PrehospWaifRefuseCause\_id) * PrehospWaifRefuseCause\_id (N, Н) - Причина отказа от госпитализации (справочник PrehospWaifRefuseCause) * EvnPS\_IsPrehospAcceptRefuse (N, У) – Отказ в подтверждении госпитализации * EvnPS\_PrehospAcceptRefuseDT (D, У) – Дата отказа в подтверждении госпитализации * EvnPS\_OutcomeDT (D, У) – Дата исхода в приемном отделении * LpuSectionProfile\_id (N,У) –Идентификатор профиля отделения; * UslugaComplex\_id(N.У) –Код посещения; |
| **Ответ** | Успешный ответ – 0 |
| **Пример** | Запрос  Ответ |

* + 1. Редактировать движение в отделение

**PUT api/EvnSection**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnSection\_id (N,O) идентификатор движения * Evn\_setDT (D,Н) Дата и время поступления * Evn\_disDT (D,У) – Дата и время выписки; * PayType\_id (N,Н) – Тип оплаты * TariffClass\_id (N,H) – Вид (класс) тарифа * LpuSection\_id (N,Н) Отделение ЛПУ * LpuSectionWard\_id (N,H) – палатная структура * MedStaffFact\_id (N,Н) – врач * Diag\_id (N,Н) – основной диагноз * DeseaseType\_id (N, У) – Характер (справочник DeseaseType), обязательный параметр, если основной диагноз (Diag\_id) отличный от Z00-Z99 и дата поступления или дата выписки (Evn\_setDT, Evn\_disDT) больше или равна 01.11.2018. * DiagSetPhase\_id (N,Н) – стадия/фаза (для основного диагноза в движении) * EvnSection\_PhaseDescr (T,H) – расшифровка (описание фазы) * Mes\_id (N,Н) – идентификатор МЭС * Mes\_sid(N,H) – КСГ * LpuSectionProfile\_id (N,Н) – Профиль * LeaveType\_id (N, У) – Исход госпитализации * CureResult\_id (N,У) ***Регионы: Астрахань, Карелия, Крым, Пермь*** – Идентификатор итога лечения. Обязателен при указании Исхода госпитализации. * EvnLeave\_UKL(N,У) – Уровень качества лечения * ResultDesease\_id(N,У) – Исход заболевания * LeaveCause\_id (N,У) – Причина (выписки, перевода) * EvnLeave\_IsAmbul(N,У) – направлен на амбулаторное лечение * Org\_oid (N,У) – Идентификатор МО для перевода * MedStaffFact\_did(N,У) – врач установивший смерть * EvnDie\_IsAnatom(N,У) –необходимость экспертизы (если смерть) * LpuUnitType\_oid(N,У) – тип стационара при переводе (если «Исход госпитализации = Переведен в стационар) * LpuSection\_oid(N, У) – Отделение переводе (если «Исход госпитализации = Переведен в стационар или Переведен на другой профильк коек) * LpuSectionBedProfile\_oid (N, H) – профиль коек (если результат госпитализации = переведен на другой профиль коек) * LpuSectionBedProfileLink\_fedid (N,Н) – Идентификатор связки между региональным профилем коек и федеральным профилем коек. * EvnSection\_IsPriem(N,Н) – Признак приемного отделения (да/нет). |
| **Ответ** | Успешный ответ – 0 |
| **Пример** | Запрос  Ответ |

* + 1. Редактирование диагноза пациенту в стационаре: сопутствующие диагнозы или осложнения

**PUT api/EvnDiagPS**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnDiagPS\_id (N,O) – идентификатор диагноза * EvnDiagPS\_setDate (D,У) дата установки * EvnDiagPS\_setTime(V,H) – время установки * DiagSetClass\_id (N,У) вид диагноза * DiagSetType\_id (N,У) Тип диагноза * Diag\_id (N,У) Диагноз * DiagSetPhase\_id (N,У) Стадия/фаза * EvnDiagPS\_PhaseDescr (T,У) Расшифровка |
| **Ответ** | Успешный ответ - 0 |

* 1. Описание методов работы с выпиской льготных рецептов
  2. 1. Получение списка выписанных льготных рецептов

**GET api/EvnReceptList**

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_pid (N, О) – Идентификатор случая-родителя (движение в КВС, посещения в ТАП) |
| **Ответ** | Успешный ответ – список случаев выписки льготных рецептов:   * Evn\_id (N, О) – Идентификатор случая выписки льготного рецепта; * Evn\_setDT (D, О) – Дата рецепта; * EvnRecept\_Num (S, Н) – Номер рецепта; * EvnRecept\_Ser (S, О) – Серия рецепта; * PrivilegeType\_id (N, О) – Тип льготы (значение справочника dbo.PrivilegeType); * DrugComplexMnn\_id (N, О) – Медикамент МНН (значение справочника dbo. DrugComplexMnn). |

* + 1. Получение информации по выписанному льготному рецепту

**GET api/EvnRecept**

|  |  |
| --- | --- |
| **Запрос** | * Evn\_id (N, Н) – Идентификатор случая выписки льготного рецепта; * Evn\_pid (N, Н) – Идентификатор случая-родителя (движение в КВС, посещения в ТАП); * Evn\_setDT (D, Н) – Дата рецепта; * EvnRecept\_Num (S, Н) – Номер рецепта |
| **Ответ** | Если не передан ни один параметр, то ошибка.  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список льготных рецептов с информацией:   * Evn\_id (N, О) – Идентификатор случая выписки льготного рецепта; * Evn\_pid (N, О) – Идентификатор случая-родителя (движение в КВС, посещения в ТАП); * ReceptForm\_id (N, О) – Форма рецепта (значение справочника dbo.ReceptForm) * ReceptType\_id (N, О) – Тип рецепта (значение справочника dbo.ReceptType); * Evn\_setDT (D, О) – Дата рецепта; * EvnRecept\_Num (S, Н) – Номер рецепта; * EvnRecept\_Ser (S, О) – Серия рецепта; * ReceptValid\_id (N, О) – Срок действия рецепта (значение справочника dbo.ReceptValid); * LpuSection\_id (N, О) – Отделение МО (значение сущности dbo.LpuSection); * MedStaffFact\_id (N, О) – Место работы врача (значение сущности dbo.MedStaffFact); * Diag\_id (N, О) – Диагноз (значение справочника dbo.Diag); * ReceptFinance\_id (N, О) – Тип финансирования (значение справочника dbo.ReceptFinance); * ReceptDiscount\_id (N, О) – Скидка (значение справочника dbo.ReceptDiscount); * EvnRecept\_is7Noz (N, О) – Признак «7 нозологий» * ~~Drug\_id (N, О) – Медикамент (значение справочника dbo.Drug);~~ * PrivilegeType\_id (N, О) – Тип льготы (значение справочника dbo.PrivilegeType); * DrugComplexMnn\_id (N, О) – Медикамент МНН (значение справочника dbo. DrugComplexMnn); * EvnRecept\_ExtempContents (S, Н) – Состав медикамента; * EvnRecept\_Kolvo (N, О) – Количество медикамента; * EvnRecept\_Signa (S, Н) – Signa. |

* + 1. Создание случая выписки льготного рецепта

**POST api/EvnRecept**

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_pid (N, О) – Идентификатор случая-родителя (движение в КВС, посещения в ТАП); * ReceptForm\_id (N, О) – Форма рецепта (значение справочника dbo.ReceptForm) * ReceptType\_id (N, О) – Тип рецепта (значение справочника dbo.ReceptType); * Evn\_setDT (D, О) – Дата рецепта; * EvnRecept\_Num (S, Н) – Номер рецепта; * EvnRecept\_Ser (S, О) – Серия рецепта; * ReceptValid\_id (N, О) – Срок действия рецепта (значение справочника dbo.ReceptValid); * LpuSection\_id (N, О) – Отделение МО (значение сущности dbo.LpuSection); * MedStaffFact\_id (N, О) – Место работы врача (значение сущности dbo.MedStaffFact); * Diag\_id (N, О) – Диагноз (значение справочника dbo.Diag); * ReceptFinance\_id (N, О) – Тип финансирования (значение справочника dbo.ReceptFinance); * ReceptDiscount\_id (N, О) – Скидка (значение справочника dbo.ReceptDiscount); * EvnRecept\_is7Noz (N, О) – Признак «7 нозологий» * PrivilegeType\_id (N, О) – Тип льготы (значение справочника dbo.PrivilegeType); * ~~Drug\_id (N, О) – Медикамент (значение справочника dbo.Drug);~~ * DrugComplexMnn\_id (N, О) – Медикамент МНН (значение справочника dbo. DrugComplexMnn); * EvnRecept\_ExtempContents (S, Н) – Состав медикамента; * EvnRecept\_Kolvo (N, О) – Количество медикамента; * EvnRecept\_Signa (S, Н) – Signa. |
| **Ответ** | Создается случай выписки льготного рецепта и формируется ответ:   * Evn\_id (N, О) – Идентификатор случая выписки льготного рецепта |

* + 1. Редактирование случая выписки льготного рецепта

**PUT api/EvnRecept**

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_id (N, О) – Идентификатор случая выписки льготного рецепта; * ReceptForm\_id (N, Н) – Форма рецепта (значение справочника dbo.ReceptForm) * ReceptType\_id (N, Н) – Тип рецепта (значение справочника dbo.ReceptType); * Evn\_setDT (D, Н) – Дата рецепта; * EvnRecept\_Num (S, Н) – Номер рецепта; * EvnRecept\_Ser (S, Н) – Серия рецепта; * ReceptValid\_id (N, Н) – Срок действия рецепта (значение справочника dbo.ReceptValid); * LpuSection\_id (N, Н) – Отделение МО (значение сущности dbo.LpuSection); * MedStaffFact\_id (N, Н) – Место работы врача (значение сущности dbo.MedStaffFact); * Diag\_id (N, Н) – Диагноз (значение справочника dbo.Diag); * ReceptFinance\_id (N, Н) – Тип финансирования (значение справочника dbo.ReceptFinance); * ReceptDiscount\_id (N, Н) – Скидка (значение справочника dbo.ReceptDiscount); * EvnRecept\_is7Noz (N, Н) – Признак «7 нозологий» * PrivilegeType\_id (N, Н) – Тип льготы (значение справочника dbo.PrivilegeType); * ~~Drug\_id (N, Н) – Медикамент (значение справочника dbo.Drug);~~ * DrugComplexMnn\_id (N, Н) – Медикамент МНН (значение справочника dbo. DrugComplexMnn); * EvnRecept\_ExtempContents (S, Н) – Состав медикамента; * EvnRecept\_Kolvo (N, Н) – Количество медикамента; * EvnRecept\_Signa (S, Н) – Signa. |
| **Ответ** | Если случай выписки льготного рецепта найден и отредактирован, то формируется успешный ответ – пустой ответ с кодом ошибки «0» |

* 1. Описание методов работы с Направлением
  2. 1. Получение списка направлений

**GET api/EvnDirectionList**

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_pid (N, Н) – Идентификатор случая-родителя (движение в КВС, посещения в ТАП); * Person\_id (N, Н) – Идентификатор человека.   Контроль: хотя бы один из входящих параметров должен быть заполнен. |
| **Ответ** | Успешный ответ – список направлений:   * EvnDirection\_id (N, О) – Идентификатор направления; * Evn\_id (N, О) – Идентификатор случая выписки направления (значение сущности dbo.Evn); * Evn\_pid (N, Н) – Идентификатор случая-родителя (движение в КВС, посещения в ТАП) (значение сущности dbo.Evn); * Person\_id (N, О) – Человек (значение сущности dbo.Person); * EvnDirection\_Num (N, О) – Номер направления; * EvnDirection\_setDate (D, О) – Дата направления; * DirType\_id (N, О) – Тип направления (значение справочника dbo.DirType) |

* + 1. Получение информации по направлению

**GET api/EvnDirection**

|  |  |
| --- | --- |
| **Запрос** | * EvnDirection\_id (N, Н) – Идентификатор направления; * Person\_id (N, Н) – Человек (значение сущности dbo.Person); * Evn\_pid (N, Н) – Идентификатор случая-родителя (движение в КВС, посещения в ТАП) (значение сущности dbo.Evn); * EvnDirection\_Num (N, Н) – Номер направления; * EvnDirection\_setDate (D, Н) – Дата направления; * DirType\_id (N, H) – Тип направления (значение справочника dbo.DirType); * Lpu\_id (N, H) – МО, куда направили * EvnDirection\_beg (DT, Н) – Дата и время начала периода изменения направления на исследование; * EvnDirection\_end (DT, Н) – Дата и время окончания периода изменения направления на исследование; |
| **Ответ** | Если не передан ни один параметр, то ошибка.  Если переданы параметры EvnDirection\_beg или EvnDirection\_end, при этом DirType\_id≠10, то ошибка: «Запрос направлений за период доступен только для направлений с типом «На исследование». Укажите корректный тип направления или удалите период. Ок»  Если в запросе указаны параметры (EvnDirection\_beg и/или EvnDirection\_end) и DirType\_id=10, то поиск осуществляется по направлениям, для которых выполняется хотя бы одно условие:   * Evn\_updDT назначенных услуг, связанных с направлением, входит в период времени, указанный в запросе; * EvnDirection\_failDT входит в период времени, указанный в запросе; * Evn\_insDT или Evn\_updDT направления входит в период времени, указанный в запросе;   Если не найдено ни одной записи, то ответ «0».  Если найдено более 10000 записей, то ошибка: «Найдено более 10000 записей, пожалуйста, уточните запрос»  Если найдено записей от 1 до 10000, то успешный ответ – список направлений с информацией:   * EvnDirection\_id (N, О) – Идентификатор направления; * Evn\_id (N, О) – Идентификатор случая выписки направления (значение сущности dbo.Evn); * Evn\_pid (N, Н) – Идентификатор случая-родителя (движение в КВС, посещения в ТАП) (значение сущности dbo.Evn); * Person\_id (N, О) – Человек (значение сущности dbo.Person); * EvnDirection\_Num (N, О) – Номер направления; * PayType\_id (N, Н) – Вид оплаты (значение справочника dbo.PayType); * DirType\_id (N, О) – Тип направления (значение справочника dbo.DirType); * EvnDirection\_setDate (D, Н) – Дата направления; * EvnDirection\_insDT (D, O) - Дата и время создания направления (Evn\_insDT) * Diag\_id (N, О) – Диагноз (значение справочника dbo.Diag); * Diag\_Code (T[10], O) – Код диагноза в МКБ-10 * EvnDirection\_Descr (S, Н) – Обоснование; * Lpu\_sid (N, О) – Направившее МО (значение справочника dbo.Lpu); * LpuSection\_id (N, О) – Направившее отделение МО (значение сущности dbo.LpuSection); * MedPersonal\_id (N, О) – Направивший врач (значение сущностиpersis.MedWorker); * MedStaffFact\_id (N, О) – Место работы направившего врача (значение сущности dbo.MedStaffFact); * MedPersonal\_zid (N, Н) – Заведующий направившего отделения (значение сущности persis.MedWorker); * Lpu\_did (N, О) – МО, куда направили (значение справочника dbo.Lpu); * LpuUnit\_did (N, О) – Группа отделений, куда направили (значение сущности dbo.LpuUnit); * LpuSectionProfil\_id (N, О) – Профиль, куда направили (значение справочника dbo.LpuSectionProfil); * MedPersonal\_did (N, Н) – Врач, к кому направили (значение сущности persis.MedWorker) * TimeTableStac\_id (N, H) – Идентификатор бирки по стационару; * DirFailType\_id (N, H) – Идентификатор типа отказа в госпитализации; * DirFailType\_Name (T[50], H) – Наименование типа отказа в госпитализации; * EvnDirection\_failDT (D, H) – Дата отмены направления; * pmUser\_failID (N, H) – Идентификатор пользователя, отказавшего в госпитализации; * TimeTableGraf\_id (N, H) – Идентификатор поликлинической бирки; * TimeTableStac\_id (N, H) – Идентификатор стационарной бирки; * TimeTableMedService\_id (N, H) – Идентификатор лабораторной бирки; * TimeTableResource\_id (N, H) – Идентификатор диагностической бирки; * EvnQueue\_id (N, H) – Идентификатор постановки в очередь. * EvnPrescr\_id (N, H) – Идентификатор назначения. * PrescriptionType\_id (N, H) – Тип назначения (значение справочника dbo.PrescriptionType). * EvnPrescr\_IsCito (N, H) – Признак «Cito» (0 – Нет, 1 – Да). * UslugaComplex\_id (N, H) – Идентификатор услуги (значение справочника dbo.UslugaComplex). * UslugaComplexParList (T, У) – Список исследований (включая идентификаторы событий оказания услуг данных исследований) и тестов параклинических услуг направления. Передается только, если тип направления «На исследование» (DirType\_id = 10). Передается JSON-строка с перечнем исследований направления, идентификаторами событий оказания услуг данных исследований и тестами данных исследований. Формат строки:   + {Evn\_id: id события оказания услуги1,   + UslugaComplex\_id: «id услуги исследования1»,   + UslugaList: [«id теста1 исследования1», «id теста2 исследования1»,…]} * Resource\_id (N, H) – Идентификатор ресурса. |
| **Пример** |  |

* + 1. Создание направления

**POST api/EvnDirection**

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_pid (N, Н) – Идентификатор случая-родителя (движение в КВС, посещения в ТАП) (значение сущности dbo.Evn); * Person\_id (N, О) – Человек (значение сущности dbo.Person); * EvnDirection\_Num (N, О) – Номер направления; * EvnDirection\_setDate (D, О) – Дата направления; * PayType\_id (N, Н) – Вид оплаты (значение справочника dbo.PayType); * DirType\_id (N, О) – Тип направления (значение справочника dbo.DirType); * Diag\_id (N, О) – Диагноз (значение справочника dbo.Diag); * EvnDirection\_Descr (S, Н) – Обоснование; * Lpu\_sid (N, О) – Направившее МО (значение справочника dbo.Lpu); * LpuSection\_id (N, O) – Направившее отделение МО (значение сущности dbo.LpuSection); * MedPersonal\_id (N, O) – Направивший врач (значение сущностиpersis.MedWorker); * MedStaffFact\_id (N, O) – Место работы направившего врача (значение сущности dbo.MedStaffFact); * MedPersonal\_zid (N, Н) – Заведующий направившего отделения (значение сущности persis.MedWorker); * Lpu\_did (N, О) – МО, куда направили (значение справочника dbo.Lpu); * LpuUnit\_did (N, O) – Группа отделений, куда направили (значение сущности dbo.LpuUnit); * LpuSectionProfile\_id (N, O) – Профиль, куда направили (значение справочника dbo.LpuSectionProfile); * MedPersonal\_did (N, Н) – Врач, к кому направили (значение сущности persis.MedWorker); * TimeTableGraf\_id (N, H) – Идентификатор поликлинической бирки; * TimeTableStac\_id (N, H) – Идентификатор стационарной бирки; * TimeTableMedService\_id (N, H) – Идентификатор лабораторной бирки; * TimeTableResource\_id (N, H) – Идентификатор диагностической бирки. * PrescriptionType\_id (N, У) – Тип назначения (значение справочника dbo.PrescriptionType). Может принимать значение 11- лабораторная диагностика, 12 – инструментальная диагностика, 6 – манипуляции и процедуры, 13 – консультационная услуга, 14 – операционный блок. Обязательный параметр, если DirType\_id = 10 (на исследование), 11 (на консультацию), 15 (на процедуры), 20 (в операционный блок). * EvnPrescr\_IsCito (N, У) – Признак «Cito» (0 – Нет, 1 – Да). Обязательный параметр, если DirType\_id = 10, 11, 15 или 20; * UslugaComplex\_id (N, У) – Идентификатор услуги (значение справочника dbo.UslugaComplex). Обязательный параметр, если DirType\_id = 10, 11, 15 или 20; * MedService\_id (N, H) – Идентификатор службы. * Resource\_id (N, H) – Идентификатор ресурса.[#99827](https://redmine.swan.perm.ru/issues/99827) |
| **Ответ** | Создается направление и формируется ответ, причем, если во входящих параметрах нет идентификатора какой-либо бирки, то вызываем метод добавления записи в лист ожидания POST api/EvnQueue/EvnQueue и в ответе получаем обязательный параметр EvnQueue\_id:   * EvnDirection\_id (N, О) – Идентификатор направления; * Evn\_id (N, О) – Идентификатор случая выписки направления (значение сущности dbo.Evn) * EvnQueue\_id (N, Y) – Идентификатор постановки в очередь.   Если входящий параметр DirType\_id = 10, 11, 15 или 20, то создается назначение, связанное с направлением и таким же случаем-родителем Evn\_pid, в ответе получаем обязательный параметр:   * EvnPrescr\_id (N, У) – Идентификатор назначения.   При создании направления предусмотрено сохранение RecMethodType\_id = «РИШ» |
| **Пример** |  |

* + 1. Отмена направления

**PUT api/EvnDirectionCancel**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnDirection\_id (N, О) – Идентификатор направления. * EvnStatusCause\_id (N, Н) – Идентификатор причины установки статуса. * EvnStatusHistory\_Cause (T, Н) – Комментарий к причине отмены направления. |
| **Ответ** | При отмене направления освобождается занятая бирка или запись в очереди, отменяется назначение, связанное с направлением, статус направления = «Отменено», в историю изменения статусов записывается причина отмены и комментарий к причине отмены.  Если направление найдено и отменено, то формируется успешный ответ – пустой ответ с кодом ошибки «0». |

* + 1. Получение атрибутов стационарной бирки по идентификатору

**GET api/TimeTableStacById**

|  |  |
| --- | --- |
| **Запрос** | * TimeTableStac\_id (N, O) – идентификатор бирки по стационару |
| **Ответ** | Успешный ответ:   * EvnDirection\_id (N, H) – Идентификатор направления; * Evn\_id (N, H) – Идентификатор случая выписки направления (значение сущности dbo.Evn); * Evn\_pid (N, Н) – Идентификатор случая-родителя (движение в КВС, посещения в ТАП) (значение сущности dbo.Evn); * Person\_id (N, H) – Человек (значение сущности dbo.Person); * LpuSectionBedType\_id (N, О) – Идентификатор типа койки; * TimeTableStac\_setDate (D, О) – Дата начала госпитализации; * TimeTableType\_id (N, H) – Идентификатор типа бирки; * LpuSection\_id (N, О) – Идентификатор отделения МО |
| **Пример** |  |

* + 1. Получение атрибутов бирки на службу по идентификатору

**GET api/TimeTableMedServiceById**

|  |  |
| --- | --- |
| **Запрос** | * TimeTableMedService\_id (N, O) - Идентификатор бирки |
| **Ответ** | Успешный ответ   * TimeTableMedService\_begTime (DT, O) – Дата и время приема; * MedService\_id (N, H) – Идентификатор службы; * TimeTableType\_id (N, H) – идентификатор типа бирки. dbo.TimeTableType * TimetableMedService\_Time (N, O) – Длительность приема; * EvnDirection\_id (N, H) – Идентификатор направления; * Person\_id (N, H) – Идентификатор пациента; * TimeTableMedService\_IsDop (N, H) – признак дополнительной бирки; * TimeTableMedService\_factTime (DT, H) – Дата и время фактического приема. |

* + 1. Получение атрибутов бирки на ресурс по идентификатору

**GET api/TimeTableResourceById**

|  |  |
| --- | --- |
| **Запрос** | * TimeTableResource\_id (N, O) - Идентификатор бирки |
| **Ответ** | Успешный ответ   * TimeTableResource\_begTime (DT, O) – Дата и время приема; * Resource\_id (N, H) – Идентификатор ресурса; * TimeTableType\_id (N, H) – идентификатор типа бирки. dbo.TimeTableType * TimetableResource\_Time (N, O) – Длительность приема; * EvnDirection\_id (N, H) – Идентификатор направления; * Person\_id (N, H) – Идентификатор пациента; * TimetableResource\_Day (N, O) – День приема по бирке. * TimeTableResource\_IsDop (N, H) – признак дополнительной бирки. |

* 1. Описание методов работы с назначениями лекарственных средств
  2. 1. Получение назначений лекарственных средств

**GET api/EvnPrescrTreat**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnPrescr\_id (N, Н) – Идентификатор назначения; * EvnPrescrTreat\_id (N, Н) – Идентификатор назначения лекарственного средства; * Evn\_pid (N, Н) – Идентификатор события-родителя (Посещение ТАП, Движение КВС); * Evn\_rid (N, Н) – Идентификатор события-родителя (ТАП, КВС); * EvnClass\_id (N, Н) – Класс события (значение справочника dbo. EvnClass); * Evn\_setDT (D, Н) – Дата назначения; * DrugComplexMnn\_id (N, Н) – Медикамент (МНН). |
| **Ответ** | Если не передан ни один параметр, то ошибка.  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список назначений лекарственных средств с информацией:   * EvnPrescr\_id (N, О) – Идентификатор назначения; * EvnPrescrTreat\_id (N, О) – Идентификатор назначения лекарственного средства; * Evn\_id (N, О) – Идентификатор события; * Evn\_pid (N, О) – Идентификатор события-родителя (Посещение ТАП, Движение КВС); * Evn\_rid (N, О) – Идентификатор события-родителя (ТАП, КВС); * EvnClass\_id (N, О) – Класс события (значение справочника dbo.EvnClass); * Lpu\_id (N, О) – МО (значение справочника dbo.Lpu); * Evn\_setDT (D, О) – Дата назначения; * Person\_id (N, О) – Человек (значение сущность dbo.Person); * PrescriptionType\_id (N, О) – Тип назначения (значение справочника dbo.PrescriptionType); * EvnPrescr\_IsCito (N, О) – Признак "Cito" (0 – Нет, 1 - Да); * PrescriptionStatusType\_id (N, О) – Статус назначения (значение справочника dbo.PrescriptionStatusType); * EvnPrescr\_Descr (T, Н) – Комментарий к назначению; * EvnPrescr\_isExec (N, О) – Признак "Выполнено" (0 – Нет, 1 - Да); * EvnPrescrTreat\_PrescrCount (N, О) – Количество приемов в сутки; * EvnCourse\_id (N, О) – Идентификатор курса; * EvnCourseTreat\_id (N, О) – Идентификатор курса приема лекарственных средств; * CourseType\_id (N, О) – Тип курса лечения (значение справочника dbo.CourseType); * EvnCourse\_Duration (N, О) – Продолжительность приема; * DurationType\_id (N, О) – Тип продолжительности приема (значение справочника dbo.DurationType); * EvnCourse\_ContReception (N, О) – Продолжительность непрерывного приема; * DurationType\_recid (N, О) – Тип продолжительности; непрерывного приема (значение справочника dbo.DurationType); * EvnCourse\_Interval (N, Н) – Продолжительность интервала; * DurationType\_intid (N, О) – Тип продолжительности интервала (значение справочника dbo.DurationType); * PrescriptionIntroType\_id (N, О) – Способ применения (значение справочника dbo.PrescriptionIntroType); * PerformanceType\_id (N, Н) – Тип исполнения (значение справочника dbo.PerformanceType);   --- Информация по медикаменту:   * DrugComplexMnn\_id (N, О) – Медикамент (МНН) (значение справочника dbo.DrugComplexMnn); * Drug\_id (N, Н) – Медикамент (значение справочника rls.Drug); * EvnPrescrTreatDrug\_Kolvo (N, О) – Доза на 1 прием; * GoodsUnit\_id (N, О) – Единица измерения дозы на 1 прием (значение справочника dbo.GoodsUnit); * EvnPrescrTreatDrug\_KolvoEd (N, О) – Количество лекарственного средства на 1 прием; * GoodsUnit\_sid (N, О) – Единица измерения количества лекарственного средства на 1 прием (значение справочника dbo.GoodsUnit); * EvnPrescrTreatDrug\_DoseDay (N, Н) – Курсовая доза; * EvnCourseTreatDrug\_Kolvo (N, О) – Доза на 1 прием; * EvnCourseTreatDrug\_KolvoEd (N, О) – Количество лекарственного средства на 1 прием; * EvnCourseTreatDrug\_PrescrDose (N, Н) – Курсовая доза. |

* + 1. Создание назначения лекарственного средства

**POST api/EvnPrescrTreat**

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_pid (N, О) – Идентификатор события-родителя (Посещение ТАП, Движение КВС); * Evn\_rid (N, О) – Идентификатор события-родителя (ТАП, КВС); * EvnClass\_id (N, Н) – Класс события (значение справочника dbo.EvnClass); * Lpu\_id (N, О) – МО (значение справочника dbo.Lpu); * Evn\_setDT (D, О) – Дата назначения; * Person\_id (N, О) – Человек (значение сущность dbo.Person); * PrescriptionType\_id (N, О) – Тип назначения (значение справочника dbo.PrescriptionType); * EvnPrescr\_IsCito (N, О) – Признак «Cito»; где 0 – Нет, 1 – Да. * PrescriptionStatusType\_id (N, О) – Статус назначения (значение справочника dbo.PrescriptionStatusType); * EvnPrescr\_Descr (T, Н) – Комментарий к назначению; * EvnPrescr\_isExec (N, О) – Признак «Выполнено»; где 0 – Нет, 1 – Да. * EvnPrescrTreat\_PrescrCount (N, О) – Количество приемов в сутки; * CourseType\_id (N, О) – Тип курса лечения (значение справочника dbo.CourseType); * EvnCourse\_Duration (N, О) – Продолжительность приема; * DurationType\_id (N, О) – Тип продолжительности приема (значение справочника dbo.DurationType); * EvnCourse\_ContReception (N, О) – Продолжительность непрерывного приема; * DurationType\_recid (N, О) – Тип продолжительности; непрерывного приема (значение справочника dbo.DurationType); * EvnCourse\_Interval (N, Н) – Продолжительность интервала; * DurationType\_intid (N, О) – Тип продолжительности интервала (значение справочника dbo.DurationType); * PrescriptionIntroType\_id (N, О) – Способ применения (значение справочника dbo.PrescriptionIntroType); * PerformanceType\_id (N, Н) – Тип исполнения (значение справочника dbo.PerformanceType).   --- Информация по медикаменту:   * DrugComplexMnn\_id (N, О) – Медикамент (МНН) (значение справочника dbo.DrugComplexMnn); * Drug\_id (N, Н) – Медикамент (значение справочника rls.Drug); * EvnPrescrTreatDrug\_Kolvo (N, О) – Доза на 1 прием; * GoodsUnit\_id (N, О) – Единица измерения дозы на 1 прием (значение справочника dbo.GoodsUnit); * EvnPrescrTreatDrug\_KolvoEd (N, О) – Количество лекарственного средства на 1 прием; * GoodsUnit\_sid (N, О) – Единица измерения количества лекарственного средства на 1 прием (значение справочника dbo.GoodsUnit); * EvnPrescrTreatDrug\_DoseDay (N, Н) – Курсовая доза; * EvnCourseTreatDrug\_Kolvo (N, О) – Доза на 1 прием; * EvnCourseTreatDrug\_KolvoEd (N, О) – Количество лекарственного средства на 1 прием; * EvnCourseTreatDrug\_PrescrDose (N, Н) – Курсовая доза. |
| **Ответ** | Результат метода post api/EvnPrescrTreat:  1. Записи в таблицу dbo.Evn:  1) EvnClass\_id = 63. Назначение;  2) EvnClass\_id = 68. Назначение с типом лекарственное лечение;  3) EvnClass\_id = 114. Курс лекарственных средств.  2. Запись в таблицу dbo.EvnPrescr.  3. Запись в таблицу dbo.EvnPrescrTreat.  4. Запись в таблицу dbo.EvnCourse.  5. Запись в таблицу dbo.EvnCourseTreat.  Успешный ответ – информация по созданному назначению лекарственного средства:   * EvnPrescr\_id (N, О) – Идентификатор назначения; * EvnPrescrTreat\_id (N, О) – Идентификатор назначения лекарственного средства; * Evn\_id (N, О) – Идентификатор события; * EvnCourse\_id (N, О) – Идентификатор курса; * EvnCourseTreat\_id (N, О) – Идентификатор курса приема лекарственных средств |

* + 1. Редактирование назначения лекарственного средства

**PUT api/EvnPrescrTreat**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnPrescr\_id (N, Н) – Идентификатор назначения; * EvnPrescrTreat\_id (N, Н) – Идентификатор назначения лекарственного средства; * Evn\_id (N, Н) – Идентификатор события; * Evn\_pid (N, Н) – Идентификатор события-родителя (Посещение ТАП, Движение КВС); * Evn\_rid (N, Н) – Идентификатор события-родителя (ТАП, КВС); * EvnClass\_id (N, Н) – Класс события (значение справочника dbo.EvnClass); * Lpu\_id (N, Н) – МО (значение справочника dbo.Lpu); * Evn\_setDT (D, Н) – Дата назначения; * Person\_id (N, Н) – Человек (значение сущность dbo.Person); * PrescriptionType\_id (N, Н) – Тип назначения (значение справочника dbo.PrescriptionType); * EvnPrescr\_IsCito (N, Н) – Признак "Cito" (0 – Нет, 1 - Да); * PrescriptionStatusType\_id (N, Н) – Статус назначения (значение справочника dbo.PrescriptionStatusType); * EvnPrescr\_Descr (T, Н) – Комментарий к назначению; * EvnPrescr\_isExec (N, Н) – Признак "Выполнено" (0 – Нет, 1 - Да); * EvnPrescrTreat\_PrescrCount (N, Н) – Количество приемов в сутки; * EvnCourse\_id (N, Н) – Идентификатор курса; * EvnCourseTreat\_id (N, Н) – Идентификатор курса приема лекарственных средств; * CourseType\_id (N, Н) – Тип курса лечения (значение справочника dbo.CourseType); * EvnCourse\_Duration (N, Н) – Продолжительность приема; * DurationType\_id (N, Н) – Тип продолжительности приема (значение справочника dbo.DurationType); * EvnCourse\_ContReception (N, Н) – Продолжительность непрерывного приема; * DurationType\_recid (N, Н) – Тип продолжительности; непрерывного приема (значение справочника dbo.DurationType); * EvnCourse\_Interval (N, Н) – Продолжительность интервала; * DurationType\_intid (N, Н) – Тип продолжительности интервала (значение справочника dbo.DurationType); * PrescriptionIntroType\_id (N, Н) – Способ применения (значение справочника dbo.PrescriptionIntroType); * PerformanceType\_id (N, Н) – Тип исполнения (значение справочника dbo.PerformanceType).   --- Информация по медикаменту:   * DrugComplexMnn\_id (N, Н) – Медикамент (МНН) (значение справочника dbo.DrugComplexMnn); * Drug\_id (N, Н) – Медикамент (значение справочника rls.Drug); * EvnPrescrTreatDrug\_Kolvo (N, Н) – Доза на 1 прием; * GoodsUnit\_id (N, Н) – Единица измерения дозы на 1 прием (значение справочника dbo.GoodsUnit); * EvnPrescrTreatDrug\_KolvoEd (N, Н) – Количество лекарственного средства на 1 прием; * GoodsUnit\_sid (N, Н) – Единица измерения количества лекарственного средства на 1 прием (значение справочника dbo.GoodsUnit); * EvnPrescrTreatDrug\_DoseDay (N, Н) – Курсовая доза; * EvnCourseTreatDrug\_Kolvo (N, Н) – Доза на 1 прием; * EvnCourseTreatDrug\_KolvoEd (N, Н) – Количество лекарственного средства на 1 прием;   EvnCourseTreatDrug\_PrescrDose (N, Н) – Курсовая доза. |
| **Ответ** | Если случай назначения лекарственных средств найден, то он удаляется и создается новый случай с новыми параметрами. В ответе метода передается новый Ид назначения (Evn\_id, EvnPrescr\_id, EvnPrescrTreat\_id). |

* 1. Описание методов работы с результатами диагностических исследований
  2. 1. Добавление исследования, прикрепленного к услуге

**POST api/LpuPacs/EvnUslugaParAssociatedResearches**

Добавление исследования, прикрепленного к услуге

|  |  |
| --- | --- |
| **Входящие параметры** | * Study\_uid (T[100], O) – уникальный идентификатор исследования. Наименование атрибута Study Instance UID, тег: (0020,000D) * EvnUslugaPar\_id (N, O) – Идентификатор параклинической услуги * Study\_date (T[20], O) – Дата исследования * Study\_time (T[20], H) – Время исследования * Patient\_Name (T[20], H) – Имя пациента |
| **Ответ** | Успешный ответ:   * EvnUslugaParAssociatedResearches\_id (N, O) – Идентификатор исследования, привязанного к услуге |

* + 1. Изменение исследования, прикрепленного к услуге

**PUT api/LpuPacs/EvnUslugaParAssociatedResearches**

Изменение исследования, прикрепленного к услуге

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnUslugaParAssociatedResearches\_id (N, O) – Идентификатор исследования, привязанного к услуге * Study\_uid (T[100], H) – уникальный идентификатор исследования. Наименование атрибута Study Instance UID, тег: (0020,000D) * EvnUslugaPar\_id (N, H) – Идентификатор параклинической услуги * Study\_date (T[20], H) – Дата исследования * Study\_time (T[20], H) – Время исследования * Patient\_Name (T[20], H) – Имя пациента |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Удаление исследования, прикрепленного к услуге

**DELETE api/LpuPacs/EvnUslugaParAssociatedResearches**

Удаление исследования, прикрепленного к услуге

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnUslugaParAssociatedResearches\_id (N, O) – Идентификатор исследования, привязанного к услуге |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Получение результатов исследования по параклинической услуге

**GET api/LpuPacs/Study\_uidByEvnUslugaPar**

Получение результатов исследования по параклинической услуге

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnUslugaPar\_id (N, O) – Идентификатор параклинической услуги. |
| **Ответ** | Успешный ответ   * Study\_uid (T[100], H) – уникальный идентификатор исследования. Наименование атрибута Study Instance UID, тег: (0020,000D) |

* + 1. Описание обработки данных о врачах, выполняющих лабораторные исследования

Логика применяется ко всем методам по работе с лабораторными исследованиями.

Обработка данных по оказывающему услугу персоналу и место оказания услуги в методах выполняется в следующем порядке:

* Если во входных параметрах метода указано место работы врача (MedStaffFact\_id), то необходимые данные берутся с данного объекта, дальнейшая обработка не осуществляется.
* Если во входных параметрах метода указан врач (MedPersonal\_id), то необходимые данные берутся с данного объекта, дальнейшая обработка не осуществляется.
* Если учетная запись ЕЦП, под которой производится работа метода, содержит ссылку на врача (MedPersonal\_id), то необходимые данные берутся с данного объекта, дальнейшая обработка не осуществляется.
* Если не выполнено ни одно из условий, то обработка пакета не осуществляется и пользователю в ответе выдается ошибка: «Не указан врач, обработка пакета невозможна».
  + 1. Добавление факта взятия пробы

POST api/EvnLabSample

Добавление информации о факте взятия пробы по направлению.

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnDirection\_id (N, O) – Идентификатор направления на лабораторное исследование * UslugaComplexList (T, O) – Список идентфикаторов услуг, по которым взята проба, через запятую. * Lpu\_did (N, H) – МО, взявшая пробу * LpuSection\_did (N, H) – Отделение, взявшее пробу * MedStaffFact\_did (N, H) – Место работы врача, взявшего пробу * MedPersonal\_did (N, H) – Врач, взявший пробу * MedPersonal\_sdid (N, H) – Средний медперсонал, взявший пробу * MedService\_did (N, H) – Служба, в которой взята проба |
| **Ответ** | Успешный ответ   * EvnLabSample\_id (N, O) – Идентификатор пробы на лабораторное исследование |
| **Пример** |  |

При отработке метода выполняются следующие действия:

* Производится проверка статуса направления:
  + Если направление отменено, то
    - Дальнейшие действия по обработке метода не производятся.
    - Формируется ответ с ошибкой «Направление отменено МО \*Наименование отменившей МО\* по причине: \*Причина отмены направления\*».
  + Иначе, продолжаются действия по отработке метода.
* Создается новая проба, привязанная к заявке из направления.
* Взятой пробе присваивается статус «Новая» (LabSampleStatus\_id=1).
* В таблице dbo.EvnLabRequestUslugaComplex по всем связкам EvnLabRequest\_id и UslugaComplex\_id , пришедшим в пакете, текущий идентификатор пробы заменяется на идентификатор пробы, созданной при отработке метода.
  + 1. Изменение результатов взятия пробы

PUT api/EvnLabSample

Изменение информации о взятии пробы по направлению

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnLabSample\_id (N, O) – Идентификатор пробы на лабораторное исследование * DefectCauseType\_id (N, Н) – Идентификатор причины брака пробы * EvnLabSample\_Comment (Т. Н) – Комментарий * EvnLabSample\_AnalyzerDate (D, Н) – Дата и время выполнения пробы на анализаторе * EvnLabSample\_DelivDT (D, Н) – Дата и время доставки пробы * EvnLabSample\_StudyDT (D, Н) – Дата и время выполнения ислледования * LabSampleStatus\_id (N, O) – Идентификатор статуса пробы (справочное значение) * Lpu\_aid (N, H) – МО, выполнившая анализ * Lpu\_did (N, H) – МО, взявшая пробу * LpuSection\_aid (N, H) – Отделение, выполнившее анализ * LpuSection\_did (N, H) – Отделение, взявшее пробу * MedStaffFact\_aid (N, H) – Место работы врача, выполнившего анализ * MedStaffFact\_did (N, H) – Место работы врача, взявшего пробу * MedPersonal\_aid (N, H) – Врач, выполнивший анализ * MedPersonal\_did (N, H) – Врач, взявший пробу * MedPersonal\_said (N, H) – Средний медперсонал, выполнивший анализ * MedPersonal\_sdid (N, H) – Средний медперсонал, взявший пробу * MedService\_did (N, H) – Служба, в которой взята проба * MedService\_id (N, H) – Служба, в которой выполняется заявка * MedService\_sid (N, H) – Служба, забраковавшая пробу |
| **Ответ** | * Успешный ответ – пустой ответ с кодом ошибки «0» |
| **Пример** |  |

* + 1. Добавление результатов лабораторного исследования

POST api/UslugaTest

Добавление результатов лабораторных исследований

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnLabSample\_id (N, О) – Идентификатор пробы. * Lpu\_id (N, H) – Идентификатор МО, в которой выполнено исследование * MedStaffFact\_id (N, H) – Место работы врача, выполнившего тест * PayType\_id (N, H) – Вид оплаты * UslugaComplex\_id (N, О) – Услуга (значение справочника dbo.UslugaComplex) * UslugaTest\_ResultValue (T250, Н) – Результат выполнения исследования * Unit\_id (N, H) – Единица измерения (справочник) * UslugaTest\_ResultUnit (T, H) – Единица измерения (текстовое значение) * UslugaTest\_Comment (T1024, H) - Комментарий * UslugaTest\_setDT (D, Н) – Дата выполнения * UslugaTest\_ResultLower (T50, H) – Нижнее референсное значение * UslugaTest\_ResultLowerCrit (T50, H) - Нижнее критическое референсное значение * UslugaTest\_ResultUpper (T50, H) – Верхнее референсное значение * UslugaTest\_ResultUpperCrit (T50, H) - Верхнее критическое референсное значение * UslugaTest\_ResultApproved (N, Н) – Статус теста («Null» - не проводился, «1» - выполнен, «2» - одобрен) * UslugaTest\_deleted (N, О) – Признак удаления («1» - не удален, «2» - удален) * UslugaTest\_delDT (D, Н) – Дата удаления теста (обязательно для заполнения, если UslugaTest\_deleted=2) * UslugaTest\_ResultCancelReason (Т, Н) – Причина отмены результата теста * UslugaTest\_ResultAppDate (D, H) – Дата обновления значения результата выполнения |
| **Ответ** | Успешный ответ:   * UslugaTest\_id (N, O) – Идентификатор лабораторного исследования |
| **Пример** |  |

При отбработке метода НЕ происходит создание UslugaTest\_id. Результаты тестов записываются в ранее созданные при взятии пробы заготовки в таблице dbo.UslugaTest , необходимый тест для заполнения ищется по связке EvnLabSample\_id и UslugaComplex\_id.

* + 1. Изменение результатов лабораторного исследования

PUT api/UslugaTest

Добавление результатов лабораторных исследований

|  |  |
| --- | --- |
| **Входящие параметры** | * UslugaTest\_id (N, O) – Идентификатор лабораторного исследования * Lpu\_id (N, Н) – Идентификатор МО, в которой выполнено исследование * MedStaffFact\_id (N, H) – Место работы врача, выполнившего тест * PayType\_id (N, H) – Вид оплаты * UslugaTest\_ResultValue (T250, Н) – Результат выполнения исследования * Unit\_id (N, H) – Единица измерения (справочник) * UslugaTest\_ResultUnit (T, H) – Единица измерения (текстовое значение) * UslugaTest\_Comment (T1024, H) - Комментарий * UslugaTest\_setDT (D, Н) – Дата выполнения * UslugaTest\_ResultLower (T50, H) – Нижнее референсное значение * UslugaTest\_ResultLowerCrit (T50, H) - Нижнее критическое референсное значение * UslugaTest\_ResultUpper (T50, H) – Верхнее референсное значение * UslugaTest\_ResultUpperCrit (T50, H) - Верхнее критическое референсное значение * UslugaTest\_ResultApproved (N, Н) – Статус теста («Null» - не проводился, «1» - выполнен, «2» - одобрен) * UslugaTest\_deleted (N, О) – Признак удаления («1» - не удален, «2» - удален) * UslugaTest\_delDT (D, Н) – Дата удаления теста (обязательно для заполнения, если UslugaTest\_deleted=2) * UslugaTest\_ResultCancelReason (Т, Н) – Причина отмены результата теста * UslugaTest\_ResultAppDate (D, H) – Дата обновления значения результата выполнения |
| **Ответ** | * Успешный ответ – пустой ответ с кодом ошибки «0» |
| **Пример** |  |

* + 1. Добавление прикрепленных файлов лабораторного исследования

POST api/EvnMediaData

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_id (N, O) – Идентификатор события (идентификатор параклинической услуги EvnUslugaPar\_id). * File (тип: строка в формате base64, обязательное поле) – прикрепленный к исследованию PDF-файл. * EvnMediaData\_FileName (Т256, Н) - Наименование файла * MedStaffFact\_id (N, H) – Место работы врача, сформировавшего файл с результатами исследований |
| **Ответ** | Успешный ответ:   * EvnMediaData\_id (N, O) |
| **Пример** |  |

Для отображения результатов в ЭМК необходимо при добавлении к услуге медиаданных заполнять Evn\_setDT данной услуги текущей датой.

* + 1. Получение прикрепленных файлов лабораторного исследования

GET api/EvnMediaData

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_id (N, O) – Идентификатор события (идентификатор параклинической услуги EvnUslugaPar\_id). * EvnMediaData\_IsRequired (N, О) – Нужен прикрепленный файл, 2 – Да, 1 – Нет. |
| **Ответ** | Успешный ответ:   * EvnMediaData\_id (N, O). * File (тип: строка в формате base64, обязательное поле) – прикрепленный к исследованию PDF-файл. Передается, если EvnMediaData\_IsRequired = 2. * EvnMediaData\_FileName (Т256, Н) - Наименование файла. Передается, если EvnMediaData\_IsRequired = 2. |
| **Пример** |  |

* + 1. Удаление файлов, прикрепленных к исследованию

DELETE api/EvnMediaData

Удаление файла, прикрепленного к исследованию.

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnMediaData\_id (N, O) – Идентификатор PDF-файла, привязанного к исследованию |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |
| **Пример** |  |

1. Запись на прием к врачу

* 1. Описание последовательности вызова методов по созданию расписания и записи к врачу

Перед созданием расписания должна быть создана Структура МО, добавлены сотрудники, созданы места работы.

Для записи человека должен быть добавлен человек, создано расписание.

Подробное описание методов см. 5.2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Описание метода** | **Метод** | **Передаваемые параметры** | **Комментарий** |
| 1 | Получение списка специальностей в МО | GET api/MedSpecOms/MedSpecOmsByMO |  |  |
| 2 | Получение списка мест работы по специальности и МО | GET api/MedStaffFact/MedStaffFactByMO | ● MedSpecOms\_id | Для получения MedSpecOms\_id используется метод Get api/Refbook с параметрами: • Refbook\_Code = dbo.MedSpecOms • Code = MedSpecOms\_Code • Name = MedSpecOms\_Name |
| 3 | Получение свободных дат приема. | GET api/TimeTableGraf/TimeTableGrafFreeDate | ● MedStaffFact\_id  ● TimeTableGraf\_beg ● TimeTableGraf\_end | Получаем в ответе TimeTableGraf\_begTime |
| 4 | Получение свободного времени приема. | GET api/TimeTableGraf/TimeTableGrafFreeTime | ● MedStaffFact\_id ● TimeTableGraf\_begTime | Для получения даты TimeTableGraf\_begTime вызваем метод 3. Если не найдено, то ошибка. Если найдены записи, то передаем параметр TimeTableGraf\_begTime |
| 5 | Запись пациента на прием. | POST api/TimeTableGraf/TimeTableGrafWrite | ● Person\_id ● TimeTableGraf\_id | Вызваем метод поиска человека GET api/PersonList. Если получаем в ответе несколько записей - ошибка. Ели нет записей - вызываем метод POST api/Person. Вызываем метод 4. Если не найдено записей, то ошибка. Если найдены записи, то передаем параметр TimeTableGraf\_id |
| 6 | Получение записей на прием по МО | GET api/TimeTableGraf/TimeTableGrafbyMO | ● TimeTableGraf\_beg  ● TimeTableGraf\_end | Получаем в ответе Person\_id, TimeTableGraf\_id |
| 8 | Добавление расписания врача | POST api/TimeTableGraf/TimeTableGraf | ● MedStaffFact\_id ● TimeTableGraf\_begTime  ● TimeTableGraf\_Time ● TimeTableType\_id  ● TimeTableGraf\_IsDop | Сначала вызываем поиск места работы врача. Если не находим, вызываем метод POST api/MedStaffFact. Для получения TimeTableType\_id используется метод Get api/Refbook с параметрами: • Refbook\_Code = dbo.TimeTableType • Code = TimeTableType\_Code • Name = TimeTableType\_Name |
| 9 | Изменение расписания врача | PUT api/TimeTableGraf/TimeTableGraf | ● MedStaffFact\_id ● TimeTableGraf\_id  ● TimeTableType\_id ● TimeTableGrafDelStatus  ● TimeTableGraf\_IsDop |  |
| 10 | Получение статуса записи на прием | GET api/TimeTableGraf/TimeTableGrafStatus | ● Person\_id , ● TimeTableGraf\_id | Вызваем метод поиска человека GET api/PersonList. Если не найдены записи или их несколько - ошибка.  Вызываем метод 6. Если в ответе нет искомого Person\_id - ошибка. |
| 11 | Изменение статуса записи на прием | PUT api/TimeTableGraf/TimeTableGrafStatus | ● Person\_id,  ● TimeTableGraf\_id ● EvnStatus\_id |  |
| 12 | Добавление записи в лист ожидания | POST api/EvnQueue/EvnQueue | ● Person\_id (N, О) ● LpuSectionProfile\_id ● LpuSection\_id (N, O) | Вызваем метод поиска человека GET api/PersonList. Если получаем в ответе несколько записей - ошибка. Ели нет записей - вызываем метод POST api/Person. Вызываем метод поиска отделения по коду и наименованию GET api/lpu/LpuSectionListByCodeAndName. Если в ответе несколько записей, то ошибка. Если не найдено записей вызываем метод POST api/Lpu/LpuSection.  Для получения LpuSectionProfile\_id используется метод Get api/Refbook с параметрами: • Refbook\_Code = dbo.LpuSectionProfile • Code = LpuSectionProfile\_Code • Name = LpuSectionProfile\_Name |
| 13 | Изменение статуса записи в листе ожидания | PUT api/EvnQueue/EvnQueueStatus | ● EvnQueue\_id ● QueueFailCause\_id |  |
| 14 | Добавление примечания к расписанию | POST api/Annotation/Annotation | ● AnnotationType\_id  ● Annotation\_Comment  ● Annotation\_begDate  ● Annotation\_begTime  ● Annotation\_endDate  ● Annotation\_endTime  ● AnnotationVison\_id ● MedStaffFact\_id | Сначала вызываем поиск места работы врача. Если не находим, вызываем метод POST api/MedStaffFact. Для получения AnnotationType\_id используется метод Get api/Refbook с параметрами: • Refbook\_Code = dbo.AnnotationType • Code = AnnotationType\_Code • Name = AnnotationType\_Name Для получения AnnotationVision\_id используется метод Get api/Refbook с параметрами: • Refbook\_Code = dbo.AnnotationVision • Code = AnnotationVision\_Code • Name = AnnotationVision\_Name |
| 15 | Изменение примечания к расписанию | PUT api/Annotation/Annotation | * Annotation\_id * AnnotationType\_id * Annotation\_Comment * Annotation\_begDate * Annotation\_begTime * Annotation\_endDate * Annotation\_endTime * AnnotationVison\_id * MedStaffFact\_id |  |
| 16 | Добавление вызова врача на дом | POST api/HomeVisit/HomeVisit | ● Person\_id ● CallProfType\_id ● AdressHomeVisit\_id ● HomeVisitCallType\_id  ● HomeVisit\_setDT  ● HomeVisit\_Num  ● MedStaffFact\_id  ● HomeVisit\_Phone ● HomeVisitWhoCall\_id  ● HomeVisitSymptom\_id ● HomeVisitSymptomType\_id ● HomeVisit\_Comment  ● HomeVisitStatus\_id  ● HomeVisit\_LpuComment | Вызваем метод поиска человека GET api/PersonList. Если получаем в ответе несколько записей - ошибка. Ели нет записей - вызываем метод POST api/Person. Вызываем поиск места работы врача. Если не находим, вызываем метод POST api/MedStaffFact.  Вызываем метод создания адреса POST api/Address. Для получения CallProfType\_id используется метод Get api/Refbook с параметрами: • Refbook\_Code = dbo.CallProfType • Code = CallProfType\_Code • Name = CallProfType\_Name Для получения HomeVisitCallType\_id используется метод Get api/Refbook с параметрами: • Refbook\_Code = dbo.HomeVisitCallType • Code = HomeVisitCallType\_Code • Name = HomeVisitCallType\_Name Для получения HomeVisitWhoCall\_id используется метод Get api/RefbookMap с параметрами: • Refbook\_Code = 1.2.643.5.1.13.2.1.1.270 • Code = HomeVisitWhoCall\_Code • Name = HomeVisitWhoCall\_Name Для получения HomeVisitStatus\_id используется метод Get api/Refbook с параметрами: • Refbook\_Code = dbo.HomeVisitStatus • Code = HomeVisitStatus\_Code • Name = HomeVisitStatus\_Name Для получения HomeVisitSymptom\_id используется метод Get api/Refbook с параметрами: • Refbook\_Code = dbo.HomeVisitSymptom • Code = HomeVisitSymptom\_Code • Name = HomeVisitSymptom\_Name Для получения HomeVisitSymptomType\_id используется метод Get api/Refbook с параметрами: • Refbook\_Code = dbo.HomeVisitSymptomType • Code = HomeVisitSymptomType\_Code • Name = HomeVisitSymptomType\_Name |
| 17 | Изменение статуса вызова врача на дом | PUT api/HomeVisitStatus/HomeVisitStatus | ● HomeVisitStatus\_id  ● HomeVisit\_id |  |

* 1. Описание REST методов работы с расписанием
     1. Получение списка специальностей в МО

**GET api/MedSpecOms/MedSpecOmsByMO**

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N, O) - Идентификатор МО * For\_Record (N,У) – параметр для определения первой свободной даты записи. Если передан, то будет считаться количество бирок доступных для записи с типами: Обычная, Запись через инфомат, Для интернета. |
| **Ответ** | Успешный ответ - список специальностей в МО, для которых есть хотя бы одно место работы на текущую дату (MedStaffFact).   * MedSpecOms\_id (N, O) – Идентификатор специальности. * TimetableGraf\_Count (N, У) – Количество доступных бирок для записи (выгружается, если For\_Record = 1) * **Регион: Пенза** MedSpecClass\_id (N,О) – Идентификатор специальности по V021. * **Регион: Пенза** MedSpecClass\_Name (T,О) – Наименование специальности по V021. |
| **Пример** | Запрос:  <http://penza.swn.local/api/MedSpecOms/MedSpecOmsByMO?forRecord>  =1&sess\_id=g7pb894p3tdj8s3vqtfcofmum6&Lpu\_id=13003995  Ответ:  "MedSpecOms\_id": "3815", "TimetableGraf\_Count": 0 |

* + 1. Получение списка мест работы по специальности и МО

**GET api/MedStaffFact/MedStaffFactByMO**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedSpecOms\_id (N, O) - Идентификатор специальности * Lpu\_id (N, O) - Идентификатор МО |
| **Ответ** | Успешный ответ - массив объектов:   * MedStaffFact\_id (N, O) – Место работы врача (значение сущности dbo.MedStaffFact); * Person\_id (N, О) – Идентификатор человека по месту работы (находим Medpersonal\_id, по нему находим persis.medworker.id, по нему находим Person\_id):   + PersonSurName\_SurName (S[50], Н) – Фамилия;   + PersonFirName\_FirName (S[50], Н) – Имя;   + PersonSecName\_SecName (S[50], Н) – Отчество; * RecType\_id (N, H) – тип записи(ссылка на справочник RecType). * LpuUnitType\_id (N, О) – Тип подразделения МО. * MedStaffFact\_setDate (D, О) – дата начала работы сотрудника. * MedStaffFact\_disDate (D, Н) – дата окончания работы сотрудника. |
| **Пример** | Запрос:  <http://penza.swn.local/api/MedStaffFact/MedStaffFactByMO?begDate=01-01-2018&sess_id=g7pb894p3tdj8s3vqtfcofmum6&Lpu_id=13003995&MedSpecOms_id=3815>  Ответ:  "MedStaffFact\_id": "99560126751", "Person\_id": "3935742", "PersonSurName\_SurName": "ПЕТРОВ", "PersonFirName\_FirName": "АЛЕКСАНДР", "PersonSecName\_SecName": "АЛЕКСАНДРОВИЧ", "RecType\_id": "1", "LpuUnitType\_id": "13", "MedStaffFact\_setDate": "01.12.2017", "MedStaffFact\_disDate": null |

* + 1. Получение свободных дат приема

**GET api/TimeTableGraf/TimeTableGrafFreeDate**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedStaffFact\_id (N, O) – Место работы врача * TimeTableGraf\_beg (D, O) – Дата начала диапазона * TimeTableGraf\_end (D, O) – Дата окончания диапазона * TimeTableGrafFreeDate (N, H) – Параметр учета прошедшего времени (1 – да, 0 – нет). |
| **Ответ** | Успешный ответ - список свободных дат   * TimeTableGraf\_begTime (D, O) – Даты, где есть бирки со статусом «свободная». Если TimeTableGrafFreeDate=1 и свободные бирки на текущую дату отсутствуют, дата не возвращается. |

* + 1. Получение свободного времени приема

**GET api/TimeTableGraf/TimeTableGrafFreeTime**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedStaffFact\_id (N, O) – Место работы врача * TimeTableGraf\_begTime (D, O) – свободная дата приема |
| **Ответ** | Успешный ответ - массив свободных бирок на прием к врачу: {   * TimeTableGraf\_id (N, Н) – идентификатор свободной бирки. * TimeTableGraf\_begTime (DT, O) – Дата и время начала приема * TimeTableGraf\_Time (N,O) – Длительность приема   } |

* + 1. Запись пациента на прием

**POST api/TimeTableGraf/TimeTableGrafWrite**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, О) – Идентификатор пациента; * TimeTableGraf\_id (N, O) - идентификатор свободной бирки * EvnQueue\_id (N, H) – идентификатор постановки в очередь |
| **Ответ** | Успешный ответ – составной идентификатор записи на прием:   * Person\_id (N, O), TimeTableGraf\_id (N, O) – составной идентификатор записи на прием.   Если во входящих параметрах указан идентификатор постановки в очередь, то в направление предается TimeTableGraf\_id, а Evn\_Queue\_id удаляется. |

* + 1. Получение записей на прием по МО

**GET api/TimeTableGraf/TimeTableGrafbyMO**

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N, O) - Идентификатор МО * TimeTableGraf\_beg (D, O) - Дата начала диапазона * TimeTableGraf\_end (D, O) - Дата окончания диапазона |
| **Ответ** | Успешный ответ - массив объектов: {   * TimeTableGraf\_id (N, O) – идентификатор бирки. * Person\_id (N, H) – Идентификатор пациента;   } |

* + 1. Получение списка записанных в МО

**GET api/TimeTableGraf/TimeTableGrafListbyMO**

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N, O)  - Идентификатор МО * TimeTableGraf\_beg (DT, O) - Дата и время начала диапазона * TimeTableGraf\_end (DT, O) - Дата и время окончания диапазона |
| **Ответ** | Успешный ответ – список записанных к врачу за диапазон   * TimeTableGraf\_begTime (DT, O) – дата и время записи * PersonInn\_Inn (N, O) – **Регион: Казахстан** ИИН пациента (для остальных регионов ИНН пациента). Если в системе не хранится, то передается значение: двенадцать девяток "999999999999") * Post\_id (N, O) – должность врача * PersonSurName\_SurName (T[50], O) – Фамилия врача; * PersonFirName\_FirName (T[50], O) – Имя врача; * PersonSecName\_SecName (T[50], Н) – Отчество врача; |
| **Пример** | **Регион: Казахстан!:**  Запрос  Для получения идентификатора МО используем метод **GET api/lpu**  Ответ  Для получения Post\_id: по записи TimeTableGraf находим место работы MedstaffFact\_id, по нему находим должность Post\_id  Для получения ФИО: по записи TimeTableGraf находим место работы MedstaffFact\_id, по нему находим Medpersonal\_id, по нему находим persis.medworker.id, по нему находим Person\_id, по нему находим соответствующие поля ФИО.  Для получения ИИН: по записи TimeTableGraf находим человека Person\_id, в нем нужное поле. |

* + 1. Добавление расписания врача

**POST api/TimeTableGraf/TimeTableGraf**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedStaffFact\_id (N, O) – Место работы врача   Массив данных для создания бирок: TimeTableGrafCreate{   * TimeTableGraf\_begTime (DT, O) – Дата и время начала приема * TimeTableGraf\_Time (N,O) – Длительность приема * TimeTableType\_id (N, O) – идентификатор типа бирки. dbo.TimeTableType   TimeTableGraf\_IsDop (N, О) – признак дополнительной бирки. Возможные значения: 0 и 1, - где 0 – нет, 1 – да.  } |
| **Ответ** | Успешный ответ – список бирок   * TimeTableGraf\_id (N, O) - Идентификатор бирки |

* + 1. Изменение расписания врача

**PUT api/TimeTableGraf/TimeTableGraf**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedStaffFact\_id (N, O) – Место работы врача   Массив объектов: TimeTableGrafEdit {   * TimeTableGraf\_id (N, O) - Идентификатор бирки * [TimeTableType](http://fixuploader.swn.local/dbspr/index/627845649)\_id (N, H) –тип бирки, на которую нужно изменить. Значение из справочника dbo.TimeTableType * TimeTableGrafDelStatus (N, H) – признак удаления бирки * TimeTableGraf\_IsDop (N,H) – признак дополнительной бирки   } |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

* + 1. Получение атрибутов бирки по идентификатору

**GET api/TimeTableGraf/TimeTableGrafById**

|  |  |
| --- | --- |
| **Входящие параметры** | * TimeTableGraf\_id (N, O) - Идентификатор бирки |
| **Ответ** | Успешный ответ   * MedStaffFact\_id (N, O) – Место работы врача * TimeTableGraf\_begTime (DT, O) – Дата и время начала приема * TimeTableGraf\_Time (N,O) – Длительность приема * TimeTableType\_id (N, O) – идентификатор типа бирки. [dbo.TimeTableType](http://fixuploader.swn.local/dbspr/index/627845649) * TimeTableGraf\_IsDop (N, О) – признак дополнительной бирки. Возможные значения: 0 и 1, - где 0 – нет, 1 – да. |

* + 1. Получение статуса записи на прием

**GET api/TimeTableGraf/TimeTableGrafStatus**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, O), TimeTableGraf\_id (N, O) – составной идентификатор записи на прием. |
| **Ответ** | Успешный ответ – статус записи на прием:   * EvnStatus\_id (N,O) – Идентификатор статуса направления (через EvnDirection\_id) |

* + 1. Изменение статуса записи на прием

**PUT api/TimeTableGraf/TimeTableGrafStatus**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, O), TimeTableGraf\_id (N, O) – составной идентификатор записи на прием. * EvnStatus\_id (N,O) – Идентификатор статуса направления * TimeTableGraf\_id (N, O) **–** ссылка на бирку поликлиники |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0»  Если EvnStatus\_id меняется на 12 или 13, то в таблице dbo.TimeTableGraf значения полей RecClass\_id, Person\_id, EvnDirection\_id меняется на NULL. |

* + 1. Добавление записи в лист ожидания

**POST api/EvnQueue/EvnQueue**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, О) – Идентификатор пациента; * LpuSectionProfile\_id (N, У) – Идентификатор профиля отделения МО. * LpuSection\_id (N, Н) – Идентификатор отделения. * MedPersonal\_id (N, Н) – Направивший врач (значение сущности persis.MedWorker); * MedStafFact\_did (N, У) – Врач, к кому в лист ожидания включается пациент. Указывается, если запись выполняется к конкретному врачу. * Lpu\_sid (N, Н) – Направившее МО (значение справочника dbo.Lpu); * Lpu\_did (N, О) – МО, куда направили (значение справочника dbo.Lpu) * EvnQueue\_desDT (D, Н) – дата и время посещения (желаемая). |
| **Действия при получении запроса** | 1. 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. 2. Если по переданному значению Person\_id не найден пациент, то ошибка «Пациент не найден в системе». 3. Если запись с аналогичными параметрами найдена в системе, то ошибка «Данные записи не прошли проверку на дублирование». Создание записи не производится. 4. Если в любом передаваемом параметре передается значение, не соответствующее ожидаемому формату, например:  * ожидается числовое значение, а передано текстовое значение; * ожидается параметр, состоящий из 11 символов, а передан параметр, состоящий из большего или меньшего количества символов;   то, ошибка: «Неверный формат поля <Наименования полей через запятую>», создание записи в БД не происходит.   1. Если в передаваемом параметре (LpuSectionProfile\_id, LpuSection\_id, MedStafFact\_did, Lpu\_sid, Lpu\_did) передается значение id, не найденное в системе, ошибка: «Значения для <Наименования полей через запятую> не найдены в справочниках системы». Создание записи в БД не происходит.   Если есть ошибки, то возвращается ошибка.  Иначе:  – в лист ожидания добавляется запись – пациент ставится в очередь по профилю или врачу (в соответствии с полученными параметрами).  – Статус записи «в очереди» (EvnQueueStatus\_id).  – способ записи/включения в Лист ожидания - «РИШ». |
| **Ответ** | Если есть ошибки, то возвращается код ошики.  Иначе:   * EvnQueue\_id (N, O) – идентификатор постановки в очередь * EvnQueueStatus\_id (N, Н) – статус листа ожидания |
| **Пример** |  |

* + 1. Изменение статуса записи в листе ожидания

**PUT api/EvnQueue/EvnQueueStatus**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnQueue\_id (N, O) – идентификатор постановки в очередь * EvnQueueStatus\_id (N, О) – статус записи в очереди (может принимать значение Отмена). * QueueFailCause\_id (N, O) – Идентификатор причины изменения порядка в очереди. * EvnStatus\_id (N,О) – Идентификатор статуса направления. Может принимать значения 12 (отменено) или 13 (отклонено). * MedPersonal\_id (N, О) – врач, изменивший статус записи в листе ожидания (значение сущности persis.MedWorker); |
| **Действия** | 1. Выполняется проверка заполнения обязательных параметров, в т.ч. – если не заполнен ни один из условных параметров или обязательный параметр, то ошибка. 2. Если по переданному значению EvnQueue\_id не найдена запись о постановке в очередь, то ошибка «Запись о постановке в очередь с указанным EvnQueue\_id не найдена в системе». 3. Если в любом передаваемом параметре передается значение, не соответствующее ожидаемому формату, например:  * ожидается числовое значение, а передано текстовое значение; * ожидается параметр, состоящий из 11 символов, а передан параметр, состоящий из большего или меньшего количества символов;   то, ошибка: «Неверный формат поля <Наименования полей через запятую>», обновление статуса записи в БД не происходит.   1. Если в передаваемом параметре (EvnQueueStatus\_id, QueueFailCause\_id, EvnStatus\_id, MedPersonal\_id) передается значение id, не найденное в системе, ошибка: «Значения для <Наименования полей через запятую> не найдены в справочниках системы». Обновление статуса записи в БД не происходит.   Если ошибок нет, то выполняется изменение данных листа ожидания.  Если полученные параметры статус листа ожидания, статус направления отличаются от вышеуказанных возможных значений, то возвращается ошибка.  Иначе: Лист ожидания отменяется.  Если листу ожидания уже предложена запись на прием, то запись отменяется. |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |
| **Пример** |  |

* + 1. Изменение листа ожидания

**PUT api/ EvnQueue/EvnQueue**

**Изменение листа ожидания. Метод используется для отмены листа ожидания.**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnQueue\_id (N, O) – идентификатор постановки в очередь * EvnQueueStatus\_id (N, О) – статус листа ожидания |
| **Действия** | Выполняется проверка заполнения обязательных параметров, в т.ч.  – если не заполнен ни один из условных параметров или обязательный параметр, то ошибка.  Если ошибок нет, то выполняется изменение данных листа ожидания.  Ели полученных параметр статус листа оиждания не равен «Отмена», то возвращается ошибка.  Иначе: Лист ожидания отменяется.  Если листу ожидания уже предложена запись на прием, то запись отменяется, причина отмены – по инициативе пациента. |
| **Ответ** | Если есть ошибки, то возвращается код ошибки и ошибка.  Иначе: выдаются следующие данные:   * EvnQueue\_id (N, O) – идентификатор постановки в очередь Person\_id (N) – Идентификатор пациента. * Lpu\_id (N) – Идентификатор МО очереди в справочнике РИШ. * LpuSectionProfile\_id (N) – Идентификатор профиля отделения МО. * LpuSection\_id (N) – Идентификатор отделения * MedStaffFact\_id (N) – Врач, к кому в лист ожидания включен пациент (значение сущности persis.MedWorker); * EvnQueue\_desDT (D, Н) – желаемые дата и время посещения. * EvnQueue\_setDate (D) – дата постановки в очередь; * EvnQueue\_setTime (T) – время постановки в очередь * EvnQueueStatus\_id (N) – статус записи в очереди. * EvnQueue\_failDT (D) – дата отмены листа ожидания. |
| **Пример** |  |

* + 1. Добавление примечания к расписанию

**POST api/Annotation/Annotation**

|  |  |
| --- | --- |
| **Входящие параметры** | * AnnotationType\_id (N, O) - Идентификатор типа примечания. Значение из справочника dbo.AnnotationType * Annotation\_Comment (T, O) – Текст примечания; * Annotation\_begDate (D, O) – Дата начала действия примечания; * Annotation\_begTime (V, O) – Время начала действия примечания; * Annotation\_endDate (D, O) – Дата окончания действия примечания; * Annotation\_endTime (V, O) – Время окончания действия примечания; * AnnotationVison\_id (N, O) – Идентификатор видимости примечания. Значение из справочника dbo.AnnotationVison * MedStaffFact\_id (N, O) – Место работы врача |
| **Ответ** | Успешный ответ – идентификатор примечания.   * Annotation\_id (N, O) - Идентификатор примечания. |

* + 1. Изменение примечания к расписанию

**PUT api/Annotation/Annotation**

|  |  |
| --- | --- |
| **Входящие параметры** | * Annotation\_id (N, O) - Идентификатор примечания. Значение из справочника dbo.AnnotationType * AnnotationType\_id (N, H) - Идентификатор типа примечания; * Annotation\_Comment (T, H) – Текст примечания; * Annotation\_begDate (D, H) – Дата начала действия примечания; * Annotation\_begTime (V, H) – Время начала действия примечания; * Annotation\_endDate (D, H) – Дата окончания действия примечания; * Annotation\_endTime (V, H) – Время окончания действия примечания; * AnnotationVison\_id (N, H) – Идентификатор видимости примечания. Значение из справочника dbo.AnnotationVison |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0». |

* + 1. Получение листа ожидания

**GET api/ EvnQueue/EvnQueue**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, У) – Идентификатор пациента. Обязательно указывается, если не заполнены другие условные параметры; * Lpu\_id (N, У) – Идентификатор МО в справочнике РИШ. Обязательно указывается, если не указан пациент; * LpuSection\_id (N, У) – Идентификатор отделения. * LpuSectionProfile\_id (N, У) – Идентификатор профиля отделения МО. Обязательно указывается, если не указан пациент и отделение и врач. * MedStaffFact\_id (N, У) – Врач, к кому в лист ожидания включается пациент.) |
| **Действия** | Выполняется проверка заполнения обязательных параметров, в т.ч.  – если не заполнен ни один из условных параметров или обязательный параметр, то ошибка.    Если есть ошибки то, возвращается код ошибки.  Иначе:  Возвращаются записи очереди, удовлетворяющие заданным входящим параметрам  – имеют статус «в очереди»;  – или у которых направления имеют статус «Поставлено в очередь». |
| **Ответ** | Если есть ошибки, то возвращается код ошибки.  Иначе: выдаются следующие данные:   * EvnQueue\_id (N, O) – идентификатор постановки в очередь Person\_id (N) – Идентификатор пациента. * Lpu\_id (N) – Идентификатор МО очереди в справочнике РИШ. * LpuSectionProfile\_id (N) – Идентификатор профиля отделения МО. * LpuSection\_id (N) – Идентификатор отделения * MedStaffFact\_id (N) – Врач, к кому в лист ожидания включен пациент (значение сущности persis.MedWorker); * EvnQueue\_setDate (D) – дата постановки в очередь; * EvnQueue\_setTime (T) – время постановки в очередь * EvnQueueStatus\_id (N) – статус записи в очереди. * EvnQueue\_desDT (D, Н) – желаемые дата и время посещения. |
| **Пример** |  |

* 1. Описание методов работы с вызовом врача на дом

1. 3. 1. Добавление вызова врача на дом

**POST api/HomeVisit/HomeVisit**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, О) – Идентификатор пациента; * CallProfType\_id (N, O) – Идентификатор профиля вызова. Значение из справочника dbo.CallProfType * Address\_Address (T, O) – Адрес вызова; * HomeVisitCallType\_id (N, O) – Идентификатор типа вызова. Значение из справочника dbo.HomeVisitCallType * HomeVisit\_setDT (DT, O) - Дата и время вызова; * HomeVisit\_Num (T, H) - Номер вызова * MedStaffFact\_id (N, H) – Место работы врача * HomeVisit\_Phone (T, H) – Телефон обратной связи; * HomeVisitWhoCall\_id (N, O) – Идентификатор вызвавшего врача. Значение из справочника dbo.HomeVisitWhoCall * HomeVisit\_Symptoms (T, O) – Симптомы; * HomeVisit\_Comment (T, H) – Дополнительная информация; * HomeVisitStatus\_id (N, O) – Идентификатор статуса вызова. Значение из справочника dbo.HomeVisitStatus * HomeVisit\_StreetGUID (T, O) – GUID улицы из справочника ФИАС. KLAdr\_Actual = 0. Передается klstreet\_id. * HomeVisit\_House (T, O) – Номер дома; * HomeVisit\_LpuComment (T, H) – Причина отказа. Обязательное, если HomeVisitStatus = «Отказ» |
| **Ответ** | Определяется МО, участок по GUID улицы.  При возникновении ошибки передается error\_code=6 и сообщение об ошибке:   * Если МО не определена: «Не удалось определить МО, обслуживающую данный адрес». * Если участок не определен: «Не удалось определить участок, обслуживающий данный адрес».   Проверяется доступна ли служба вызова врача на дом:  Если служба вызова врача на дом недоступна: «Вызов врача на дом может быть оформлен через портал в рабочие дни поликлиники % время работы%».  Успешный ответ – идентификатор вызова на дом.   * HomeVisit\_id - Идентификатор вызова на дом. |

* + 1. Изменение статуса вызова врача на дом

**PUT api/HomeVisitStatus/HomeVisitStatus**

|  |  |
| --- | --- |
| **Входящие параметры** | * HomeVisitStatus\_id (N, O) – Идентификатор статуса вызова. Значение из справочника [dbo.HomeVisitStatus](http://fixuploader.swn.local/dbspr/index/2113103710) * HomeVisit\_id (N, O) - Идентификатор вызова на дом. |
| **Ответ** | При сохранении в поле Источники данных сохраняется значение РИШ.  Успешный ответ – пустой ответ с кодом ошибки «0». |

* + 1. Получение атрибутов вызова врача на дом по идентификатору

**GET api/HomeVisit/HomeVisitById**

|  |  |
| --- | --- |
| **Входящие параметры** | * HomeVisit\_id - Идентификатор вызова на дом. |
| **Ответ** | Успешный ответ:   * Person\_id (N, О) – Идентификатор пациента; * CallProfType\_id (N, O) – Идентификатор профиля вызова. Значение из справочника dbo.CallProfType * Address\_Address (T, O) – Адрес вызова; * HomeVisitCallType\_id (N, O) – Идентификатор типа вызова. Значение из справочника dbo.HomeVisitCallType * HomeVisit\_setDT (DT, O) - Дата и время вызова; * HomeVisit\_Num (T, O) - Номер вызова * MedStaffFact\_id (N, O) – Место работы врача * HomeVisit\_Phone (T, O) – Телефон обратной связи; * HomeVisitWhoCall\_id (N, O) – Идентификатор вызвавшего врача. Значение из справочника dbo.HomeVisitWhoCall * HomeVisit\_Symptoms (T, O) – Симптомы; * HomeVisit\_Comment (T, H) – Дополнительная информация; * HomeVisitStatus\_id (N, O) – Идентификатор статуса вызова. Значение из справочника dbo.HomeVisitStatus * HomeVisit\_LpuComment (T, H) – Причина отказа. Обязательное, если HomeVisitStatus = «Отказ» * Lpu\_Name (Т, О) – Наименование МО; * PAddress\_Address (Т, О) – Адрес МО; * Org\_Phone (Т, Н) – Телефон МО; * Person\_Fio (Т, Н) – Фамилия Имя Отчество врача. Обязательно, если статус вызова «Одобрен врачом», «Обслужен», «Назначен врач»; * MedSpecOms\_Name (Т, Н) – Специальность врача. Обязательно, если статус вызова «Одобрен врачом», «Обслужен», «Назначен врач»; |

* + 1. Отмена вызова

**PUT api/HomeVisit/HomeVisitCancel**

|  |  |
| --- | --- |
| **Входящие параметры** | * HomeVisit\_id (N, O) – Идентификатор вызова на дом. |
| **Ответ** | * HomeVisitStatus\_id (N, O) – Идентификатор статуса вызова. Значение из справочника [dbo.HomeVisitStatus](http://fixuploader.swn.local/dbspr/index/2113103710) * HomeVisitStatus\_name (Т, O) – Наименование статуса вызова. Значение из справочника [dbo.HomeVisitStatus](http://fixuploader.swn.local/dbspr/index/2113103710) |
| **Пример** |  |

* + 1. Получение симптомов вызова

**GET api/HomeVisit/HomeVisitSymptoms**

|  |  |
| --- | --- |
| **Входящие параметры** | * sess\_id – идентификатор сессии |
| **Ответ** | Массив записей с набором атрибутов:   * id (N, O) – идентификатор симптома * name (Т, O) – наименование симптома * radio (N, O) – признак радио группы * pid (N, O) – идентификатор родителя * visittype (Т, O) – тип симптомов (если HomeVisitSymptom.HomeVisitSymptomType\_id=2, то stom иначе ther) * type (Т, O) – тип элемента (если нет родительского элемента, то type=maingroup, иначе, если в родительском элементе radio<>null, то type=radio, иначе type=check. |
| **Пример** |  |

* 1. Описание методов работы с адресом
  2. 1. Получение адреса

**GET api/Address**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, Н) – Идентификатор человека; * Address\_id (N, Н) – Идентификатор адреса; * AddressType\_id (N, Н) – Тип адреса (1 - Адрес регистрации, 2 - Адрес проживания, 3 - Адрес рождения) |
| **Ответ** | Если не задан ни Person\_id, ни Address\_id, то ошибка (хотя бы один из параметров должен быть задан).  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список адресов с информацией:   * Person\_id (N, Н) – Идентификатор человека; * Address\_id (N, О) – Идентификатор адреса; * AddressType\_id (N, Н) – Тип адреса (1 - Адрес регистрации, 2 - Адрес проживания, 3 - Адрес рождения); * Address\_Zip (N, Н) – Индекс; * KLCountry\_id (N, О) – Страна (значение справочника dbo. KLCountry); * KLRgn\_id (N, О) – Регион (значение справочника dbo.Area); * KLSubRgn\_id (N, У) – Район (значение справочника dbo.Area); * KLCity\_id (N, У) – Город (значение справочника dbo.Area); * KLTown\_id (N, У) – Нас. Пункт (значение справочника dbo.Area); * KLStreet\_id (N, О) – Улица (значение справочника dbo.KLStreet); * Address\_House (N, О) – Дом; * Address\_Corpus (N, Н) – Корпус; * Address\_Flat (N, Н) – Квартира; * Address\_Address (T, Н) – Полный адрес в текстовом формате. * Address\_insDT (D, О) – Дата создания записи об адресе. * AoidArea (Т[36],У) – Идентификатор адресного объекта (населенного пункта, города) по ФИАС – значение справочника KLArea, поле KLArea\_AOID. * AoidStreet (Т[36],У) – Идентификатор адресного объекта (улицы) по ФИАС – значение справочника KLStreet, поле KLStreet\_AOID. |

* + 1. Создание адреса

**POST api/Address**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, Н) – Идентификатор человека; * AddressType\_id (N, Н) – Тип адреса (1 - Адрес регистрации, 2 - Адрес проживания, 3 - Адрес рождения); * Address\_Zip (N, Н) – Индекс; * KLCountry\_id (N, О) – Страна (значение справочника dbo. KLCountry); * KLRgn\_id (N, О) – Регион (значение справочника dbo.Area); * KLSubRgn\_id (N, У) – Район (значение справочника dbo.Area); * KLCity\_id (N, У) – Город (значение справочника dbo.Area); * KLTown\_id (N, У) – Нас. Пункт (значение справочника dbo.Area); * KLStreet\_id (N, О) – Улица (значение справочника dbo.KLStreet); * Address\_House (N, О) – Дом; * Address\_Corpus (N, Н) – Корпус; * Address\_Flat (N, Н) – Квартира. * AoidArea (Т[36],У) – Идентификатор адресного объекта (населенного пункта, города) по ФИАС – значение справочника KLArea, поле KLArea\_AOID. * AoidStreet (Т[36],У) – Идентификатор адресного объекта (улицы) по ФИАС – значение справочника KLStreet, поле KLStreet\_AOID. |
| **Выполняются действия** | Проверки:  - хотя бы один из обязательных параметров не задан, то ошибка;  - если не задан ни KLSubRgn\_id, ни KLCity\_id, ни KLTown\_id, ни AoidArea, ни AoidStreet, то ошибка (хотя бы один из параметров должен быть задан).  Если есть ошибки, то данные не создаются. Возвращается код ошибки.  Иначе: Создаются данные адреса.  Если НЕ задан Person\_id или AddressType\_id, то создается адрес без привязки к человеку.  Если задан Person\_id и AddressType\_id = 1. Адрес регистрации, то создается адрес регистрации для человека (UAddress\_id).  Если задан Person\_id и AddressType\_id = 2. Адрес проживания, то создается адрес проживания для человека (PAddress\_id).  Если задан Person\_id и AddressType\_id = 3. Адрес рождения, то создается адрес рождения для человека (BAddress\_id).   * Person\_id (N, Н) – Идентификатор человека; * Address\_id (N, О) – Идентификатор адреса;   AddressType\_id (N, Н) – Тип адреса (1 - Адрес регистрации, 2 - Адрес проживания, 3 - Адрес рождения)  **Если в параметрах получены идентификаторы адресных объектов по справочнику ФИАС** - AoidArea или AoidStreet, а идентификаторы адресных объектов по справочнику КЛАДР - Улица, Город, Населенный пункт, Район – не заданы, **то выполняется поиск адреса по идентификаторам ФИАС:**  Если задана улица - AoidStreet, то   * выполняется поиск улицы, у которой KLStreet\_AOID равен указанному параметру.   Если не найдена, то ошибка,  иначе берется KLStreet\_PGUID, и по этому ГУИД выполняется поиск в таблице KlArea находится запись, у которой KLArea\_AOGUID равен указанному значению или выполняется поиск региона, района, города, населенного пункта по полю KLArea\_AOID - выполняется поиск значения AoidArea .  В найденной записи по полю KLAreaLevel\_id  3 определяется уровень адресного объекта.  Далее: по идентификатору KLArea\_PGUID выполняется поиск адресного объекта с таким ГУИД - так определяется очередной родительский объект адреса – и т.д., пока весь адрес не будет определен. |
| **Ответ** | Если не задан ни KLSubRgn\_id, ни KLCity\_id, ни KLTown\_id, то ошибка (хотя бы один из параметров должен быть задан).  Если НЕ задан Person\_id или AddressType\_id, то создается адрес без привязки к человеку.  Если задан Person\_id и AddressType\_id = 1. Адрес регистрации, то создается адрес регистрации для человека (UAddress\_id).  Если задан Person\_id и AddressType\_id = 2. Адрес проживания, то создается адрес проживания для человека (PAddress\_id).  Если задан Person\_id и AddressType\_id = 3. Адрес рождения, то создается адрес рождения для человека (BAddress\_id).   * Person\_id (N, Н) – Идентификатор человека; * Address\_id (N, О) – Идентификатор адреса; * AddressType\_id (N, Н) – Тип адреса (1 - Адрес регистрации, 2 - Адрес проживания, 3 - Адрес рождения) |

* + 1. Редактирование адреса

**PUT api/Address**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, Н) – Идентификатор человека; * Address\_id (N, О) – Идентификатор адреса; * AddressType\_id (N, Н) – Тип адреса (1 - Адрес регистрации, 2 - Адрес проживания, 3 - Адрес рождения); * Address\_Zip (N, Н) – Индекс; * KLCountry\_id (N, Н) – Страна (значение справочника dbo. KLCountry); * KLRgn\_id (N, Н) – Регион (значение справочника dbo.Area); * KLSubRgn\_id (N, Н) – Район (значение справочника dbo.Area); * KLCity\_id (N, Н) – Город (значение справочника dbo.Area); * KLTown\_id (N, Н) – Нас. Пункт (значение справочника dbo.Area); * KLStreet\_id (N, Н) – Улица (значение справочника dbo.KLStreet); * Address\_House (N, Н) – Дом; * Address\_Corpus (N, Н) – Корпус; * Address\_Flat (N, Н) – Квартира. * AoidArea (Т[36],У) – Идентификатор адресного объекта (населенного пункта, города) по ФИАС – значение справочника KLArea, поле KLArea\_AOID. * AoidStreet (Т[36],У) – Идентификатор адресного объекта (улицы) по ФИАС – значение справочника KLStreet, поле KLStreet\_AOID. |
| **Ответ** | Если не задан ни KLSubRgn\_id, ни KLCity\_id, ни KLTown\_id, то ошибка (хотя бы один из параметров должен быть задан).  Если НЕ задан Person\_id или AddressType\_id, то редактируется адрес без привязки к человеку.  Успешный ответ – пустой ответ с кодом ошибки «0». |

1. Описание методов работы с человеком
2. 1. Получение списка физических лиц, по ключевым параметрам

**GET api/PersonList**

Получение списка пациентов по ключевым параметрам (Фамилия, Имя, Отчество, Дата рождения, СНИЛС, Серия и номер полиса).

|  |  |
| --- | --- |
| **Входящие параметры** | * PersonSurName\_SurName (S[50], О) – Фамилия; * PersonFirName\_FirName (S[50], Н) – Имя; * PersonSecName\_SecName (S[50], Н) – Отчество; * PersonBirthDay\_BirthDay (D, О) – Дата рождения; * PersonSnils\_Snils (N[11],Н) – СНИЛС; * Polis\_Ser (S[10], Н) – Серия полиса; * Polis\_Num (N[15], Н) – Номер полиса |
| **Ответ** | Успешный ответ – список пациентов с указанием следующих параметров:   * Person\_id (N, О) – идентификатор пациента; * PersonSurName\_SurName (S[50], О) – Фамилия; * PersonFirName\_FirName (S[50], Н) – Имя; * PersonSecName\_SecName (S[50], Н) – Отчество; * PersonBirthDay\_BirthDay (D, О) – Дата рождения; * PersonSnils\_Snils (N[11],Н) – СНИЛС; * Polis\_id (N, Н) – Полис:   o   PolisType\_id (N, О) – Тип полиса (значение справочника dbo.PolisType);  o   Polis\_Ser (S[10], Н) – Серия полиса;  o   Polis\_Num (N[15], О) – Номер полиса; |

* 1. Получение информации по человеку

**GET api/Person**

Получение информации по пациенту.

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, Н) – идентификатор пациента; * PersonSurName\_SurName (S[50], Н) – Фамилия; * PersonFirName\_FirName (S[50], Н) – Имя; * PersonSecName\_SecName (S[50], Н) – Отчество; * PersonBirthDay\_BirthDay (D, Н) – Дата рождения; * PersonSnils\_Snils (N[11],Н) – СНИЛС; * Polis\_Ser (S[10], Н) – Серия полиса; * Polis\_Num (N[15], Н) – Номер полиса |
| **Ответ** | Если не передан ни один параметр, то ошибка.  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список людей с информацией:   * Person\_id (N, О) – идентификатор пациента; * PersonSurName\_SurName (S[50], О) – Фамилия; * PersonFirName\_FirName (S[50], Н) – Имя; * PersonSecName\_SecName (S[50], Н) – Отчество; * PersonBirthDay\_BirthDay (D, Н) – Дата рождения; * Person\_Sex\_id (N, Н) – Пол (значение справочника dbo.sex) * PersonPhone\_Phone (S, Н) – Телефон; * PersonSnils\_Snils (N[11],Н) – СНИЛС; * PersonInn\_Inn (N, H) – ИНН пациента; * SocStatus\_id (N, Н) – Социальный статус (значение справочника dbo.SocStatus); * UAddress\_id (N, Н) – Адрес регистрации (значение сущности dbo.Address); * PAddress\_id (N, Н) – Адрес проживания (значение сущности dbo.Address); * BAddress\_id (N, Н) – Адрес рождения (значение сущности dbo.Address); * Org\_id (N, Н) – Место работы (значение справочника dbo.Org); * Post\_id (N, Н) – Должность (значение справочника dbo.Post) * BDZ\_guid (N, H) – Гуид по БДЗ; * BDZ\_id (N, Н) – Ид по БДЗ. |

* 1. Создание человека

**POST api/Person**

Создание человека.

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_isUnknown (N, Н) – признак «Личность неизвестна»; * PersonSurName\_SurName (S[50], О) – Фамилия; * PersonFirName\_FirName (S[50], Н) – Имя; * PersonSecName\_SecName (S[50], Н) – Отчество; * PersonBirthDay\_BirthDay (D, О) – Дата рождения; * Person\_Sex\_id (N, О) – Пол (значение справочника dbo.sex) * PersonPhone\_Phone (N[10], Н) – Телефон; * PersonSnils\_Snils (N[11],Н) – СНИЛС; * SocStatus\_id (N, О) – Социальный статус (значение справочника dbo.SocStatus); * UAddress\_id (N, Н) – Адрес регистрации (значение сущности dbo.Address); * PAddress\_id (N, Н) – Адрес проживания (значение сущности dbo.Address); * BAddress\_id (N, Н) – Адрес рождения (значение сущности dbo.Address); * Org\_id (N, Н) – Место работы (значение справочника dbo.Org); * Post\_id (N, Н) – Должность (значение справочника dbo.Post). * PersonInn\_Inn (N, H) – ИНН |
| **Ответ** | Алгоритм выполнения метода:   1. 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.  1. Производится 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. 2. Производится проверка: параметр PersonPhone\_Phone может содержать только 10-значное число. Если результат проверки отрицательный, то выводится ошибка о неверном формате входного параметра; 3. 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.   1. Если переданная дата рождения больше текущей даты, то ошибка «Некорректное значение в поле PersonBirthDay\_BirthDay». Создание человека не происходит. 2. 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. * Производится проверка: параметр СНИЛС может содержать только 11значный номер, состоящий из цифр. Если результат проверки отрицательный, то выводится ошибка о неверном формате входного параметра;  1. Если в передаваемом параметре (Person\_Sex\_id, SocStatus\_id, Address\_id, PAddress\_id, BAddress\_id, Org\_id, Post\_id) передается значение id, не найденное в системе, ошибка: «Значения для <Наименования полей через запятую> не найдены в справочниках системы», создание человека в БД не происходит. Аналогичный контроль производится для параметров, по которым ожидается значение 0/1 (да/нет) (Person\_isUnknown). Если контроль не пройден, то ошибка «Значение поля <названия полей через запятую> не соответствует ожидаемому», создание человека в БД не происходит. 2. 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, производится проверка на контрольную сумму. Если проверка не пройдена, создание Человека не происходит, ошибка «Ошибка проверки контрольной суммы в ИНН».   Иначе новый человек добавляется в БД. В ответе выводится информация о человеке:   * Person\_id (N, О) – идентификатор пациента |
| **Пример** |  |

* 1. Редактирование человека

**PUT api/Person**

Редактирование информации по человеку.

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, О) – идентификатор пациента; * PersonSurName\_SurName (S[50], Н) – Фамилия; * PersonFirName\_FirName (S[50], Н) – Имя; * PersonSecName\_SecName (S[50], Н) – Отчество; * PersonBirthDay\_BirthDay (D, Н) – Дата рождения; * Person\_Sex\_id (N, Н) – Пол (значение справочника dbo.sex) * PersonPhone\_Phone (N[10], Н) – Телефон; * PersonSnils\_Snils (N[11],Н) – СНИЛС; * SocStatus\_id (N, Н) – Социальный статус (значение справочника dbo.SocStatus); * UAddress\_id (N, Н) – Адрес регистрации (значение сущности dbo.Address); * PAddress\_id (N, Н) – Адрес проживания (значение сущности dbo.Address); * BAddress\_id (N, Н) – Адрес рождения (значение сущности dbo.Address); * Org\_id (N, Н) – Место работы (значение справочника dbo.Org); * Post\_id (N, Н) – Должность (значение справочника dbo.Post). * [**#143524**](https://redmine.swan.perm.ru/issues/143524)PersonInn\_Inn (N, H) – ИНН |
| **Ответ** | 1. Производится проверка: параметр PersonPhone\_Phone может содержать только 10-значное число. Если результат проверки отрицательный, то выводится ошибка о неверном формате входного параметра; 2. Если по переданному значению Person\_id не найден пациент, то ошибка «Пациент не найден в системе». 3. Если в любом передаваемом параметре передается значение, не соответствующее ожидаемому формату, например:  * ожидается числовое значение, а передано текстовое значение; * ожидается параметр, состоящий из 11 символов, а передан параметр, состоящий из большего или меньшего количества символов;   то, ошибка: «Неверный формат поля <Наименования полей через запятую>». Обновление человека в БД не происходит.   1. Производится 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. 2. Если в любом передаваемом параметре передается значение id (Person\_Sex\_id, SocStatus\_id, Address\_id, PAddress\_id, BAddress\_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.   Успешный ответ – пустой ответ с кодом ошибки «0» |
| **Пример** |  |

* 1. Получение полисных данных пациента

**GET api/Polis**

Получение информации по полису пациента.

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, Н) – идентификатор пациента; * Polis\_id (N, Н) – идентификатор полиса; * Polis\_Ser (S[10], Н) – Серия полиса; * Polis\_Num (N[15], Н) – Номер полиса. |
| **Ответ** | Если не передан ни один параметр, то метод возвращает ошибку (error\_code: 6, error\_msg: «Не передан ни один из параметров поиска»).  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – 1 полис: либо по указанному ID (серии, номеру), либо последний введенный у пациента (по Person\_id) с информацией:   * Person\_id (N, О) – идентификатор пациента; * Polis\_id (N, О) – идентификатор полиса; * OmsSpeTerr\_id (N, О) – Территория страхования (значение справочника dbo.OmsSprTerr); * PolisType\_id (N, О) – Тип полиса (значение справочника dbo.PolisType); * Polis\_Ser (S[10], Н) – Серия полиса; * Polis\_Num (N[15], О) – Номер полиса; * OrgSmoid (N, О) – СМО (значение сущности dbo.OrgSmo); * Polis\_BegDate (D, О) – Дата выдачи; * Polis\_EndDate (D, Н) – Дата закрытия; * PolisFormType\_id (N, Н) – Форма полиса (значение справочника dbo.PolisFormType) |

* 1. Создание полиса

**POST api/Polis**

Создание полиса.

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, О) – идентификатор пациента; * OmsSpeTerr\_id (N, О) – Территория страхования (значение справочника dbo.OmsSprTerr); * PolisType\_id (N, О) – Тип полиса (значение справочника dbo.PolisType); * Polis\_Ser (S[10], Н) – Серия полиса; * Polis\_Num (N[15], О) – Номер полиса; * OrgSmoid (N, О) – СМО (значение сущности dbo.OrgSmo); * Polis\_BegDate (D, О) – Дата выдачи; * Polis\_EndDate (D, Н) – Дата закрытия; * PolisFormType\_id (N, Н) – Форма полиса (значение справочника dbo.PolisFormType) |
| **Ответ** | Производятся проверки:   1. Проверка поля Polis\_Num:  * Если передан тип полиса = «1. ОМС (старого образца)» и регион системы = территории страхования, то в поле Polis\_Num доступны следующие символы: цифры (0-9). * Если передан тип полиса = «1. ОМС (старого образца)» и регион системы <> территории страхования, то в поле Polis\_Num доступны следующие символы: цифры (0-9), точка, символ «/».   Если проверка не пройдена, ошибка: «Неверный формат поля Polis\_Num». Создание полиса не происходит.   1. Если по переданному значению Person\_id не найден пациент, то ошибка «Пациент не найден в системе». 2. 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 полиса не происходит.   1. Если передан тип полиса PolisType\_id=3 (Временное свидетельство):  * Если не передано никакое значение в поле «Серия», то если длина переданного номера (Polis\_Num) составляет 9 символов, то первые 3 символа необходимо сохранить в БД в поле «Серия», в поле «Номер» сохранить последние 6 символов. * Если в поле «Серия» передано ненулевое значение, состоящее больше/меньше чем из трех цифр, то ошибка «Неверный формат данных в поле Polis\_Ser для переданного PolisType». Создание полиса не происходит. * Если и в поле «Серия» и в поле «Номер» переданы ненулевые значения, то происходит проверка: в поле «Серия» должно быть три цифры, в поле «Номер» должно быть 6 цифр. Если проверка не пройдена, то ошибка «Неверный формат поля Polis\_Ser, Polis\_Num». Создание полиса не происходит.  1. Если в передаваемом параметре (OmsSpeTerr\_id, PolisType\_id, OrgSmoid, PolisFormType\_id) передается значение id, не найденное в системе, ошибка: «Значения для <Наименования полей через запятую> не найдены в справочниках системы», создание полиса не происходит. 2. Если полис с аналогичными передаваемыми параметрами для данного человека найден в системе, создание полиса не происходит, ошибка «Данные полиса не прошли проверку на дублирование.».   Успешный ответ:   * Person\_id (N, Н) – идентификатор пациента; * Polis\_id (N, О) – идентификатор полиса |
| **Пример** |  |

* 1. Изменение полисных данных человека

**PUT api/Polis**

Редактирование информации по полису человека.

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, Н) – идентификатор пациента; * Polis\_id (N, О) – идентификатор полиса * OmsSpeTerr\_id (N, Н) – Территория страхования (значение справочника dbo.OmsSprTerr); * PolisType\_id (N, Н) – Тип полиса (значение справочника dbo.PolisType); * Polis\_Ser (S[10], Н) – Серия полиса; * Polis\_Num (N[15], Н) – Номер полиса; * OrgSmoid (N, Н) – СМО (значение сущности dbo.OrgSmo); * Polis\_BegDate (D, Н) – Дата выдачи; * Polis\_EndDate (D, Н) – Дата закрытия; * PolisFormType\_id (N, Н) – Форма полиса (значение справочника dbo.PolisFormType) |
| **Ответ** | Производятся проверки:   1. Если в любом передаваемом параметре передается значение, не соответствующее ожидаемому формату, например:  * ожидается числовое значение, а передано текстовое значение; * ожидается параметр, состоящий из 11 символов, а передан параметр, состоящий из большего или меньшего количества символов;   то, ошибка: «Неверный формат поля <Наименования полей через запятую>». Обновление данных полиса не происходит.   1. Если по переданному значению Person\_id не найден пациент, то ошибка «Пациент не найден в системе». 2. Если для пациента (Person\_id) не найден полис (Polis\_id), то ошибка «Нет данных о полисе пациента с указанным Polis\_id». 3. Дата закрытия полиса не может быть меньше даты открытия полиса (сравниваются даты открытия и закрытия с учетом полученных в методе значений). Если проверка не пройдена, то ошибка «Дата закрытия полиса не может быть больше даты открытия полиса». Обновления данных полиса не происходит. 4. Дата открытия полиса не может быть больше текущей даты (сравнивается текущая дата и полученная в методе дата открытия). Если проверка не пройдена, то ошибка «Дата открытия полиса не может быть больше текущей даты». Обновления данных полиса не происходит. 5. Если в передаваемом параметре (OmsSpeTerr\_id, PolisType\_id, OrgSmoid, PolisFormType\_id) передается значение id, не найденное в системе, ошибка: «Значения для <Наименования полей через запятую> не найдены в справочниках системы», обновление полиса не происходит. 6. Если с учетом передаваемых изменений по параметрам полис с аналогичными параметрами для данного человека найден в системе, обновление полиса не происходит, ошибка «Данные полиса не прошли проверку на дублирование.».   Если найден один человек и пройдены все проверки, то редактируются данные полиса и формируется успешный ответ – пустой ответ с кодом ошибки «0» |
| **Пример** |  |

* 1. Получение документа удостоверяющего личность

**GET api/Document**

Получение информации по ДУЛ пациента.

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, Н) – идентификатор пациента; * Document\_id (N, Н) – идентификатор ДУЛ; * Document\_Ser (S, Н) – Серия документа; * Document\_Num (S, Н) – Номер документа. |
| **Ответ** | Если не передан ни один параметр, то ошибка.  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список документов, удостоверяющих личность, с информацией:   * Person\_id (N, О) – идентификатор пациента; * Document\_id (N, О) – идентификатор ДУЛ; * DocumentType\_id (N, О) – Тип документа (значение справочника dbo.DocumentType); * Document\_Ser (S, Н) – Серия документа; * Document\_Num (S, О) – Номер документа; * OrgDep\_id (N, Н) – Организация выдачи документа (значение справочника dbo.OrgDep); * Document\_begDate (D, Н) – Дата выдачи; * KLCountry\_id (N, Н) – Гражданство (значение справочника dbo.KLCountry) * Document\_IsTwoNation (N, Н) – Признак «Двойное гражданство» (0 – Нет, 1 - Да). |

* 1. Создание документа

**POST api/Document**

Создание документа, удостоверяющего личность, на человека.

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, Н) – идентификатор пациента * DocumentType\_id (N, О) – Тип документа (значение справочника dbo.DocumentType); * Document\_Ser (S, О) – Серия документа; * Document\_Num (S, О) – Номер документа; * OrgDep\_id (N, Н) – Организация выдачи документа (значение справочника dbo.OrgDep); * Document\_begDate (D, О) – Дата выдачи; * KLCountry\_id (N, Н) – Гражданство (значение справочника dbo.KLCountry); * Document\_IsTwoNation (N, Н) – Признак «Двойное гражданство» (0 – Нет, 1 - Да). |
| **Ответ** | При одновременном выполнении условий:   1. Серия документа (Document\_Ser) и номер документа (Document\_Num) прошли валидацию (для проверки используются регулярные выражения в таблице dbo.DocumentType: DocumentType\_MaskSer, DocumentType\_MaskNum) 2. найден один человек,   И выполнении следующих проверок:   1. Если документ с аналогичными передаваемыми параметрами для данного человека найден в системе, создание документа не происходит, ошибка «Данные документа не прошли проверку на дублирование.», создание документа не производится. 2. Если по переданному значению Person\_id не найден пациент, то ошибка «Пациент не найден в системе». 3. 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 документа не производится.   1. Если в передаваемом параметре (DocumentType\_id, OrgDep\_id, KLCountry\_id) передается значение id, не найденное в системе, ошибка: «Значения для <Наименования полей через запятую> не найдены в справочниках системы». Создание документа не происходит. Аналогичный контроль производится для параметров, по которым ожидается значение 0/1 (да/нет) (Document\_IsTwoNation). Если контроль не пройден, то ошибка «Значение поля <названия полей через запятую> не соответствует ожидаемому», создание документа не производится. 2. на дату выдачи в зависимости от типа документа. Если в поле «Тип» выбран любой тип, **кроме**:  * DocumentType\_id = 17 иные документы, * DocumentType\_id = 3 свидетельство о рождении РФ, * DocumentType\_id = 9 паспорт иностранного гражданина, * DocumentType\_id = 19 документ иностранного гражданина, * DocumentType\_id = 22 свидетельство о рождении, выданное не в РФ   то выполняется проверка: переданная в поле Document\_begDate «Дата выдачи» дата должна быть больше или равна «Дата рождения» человека Person\_id + 14 лет. Если условие не выполняется, то пользователю выводится сообщение: «Дата выдачи документа Document\_begDate должна соответствовать дате 14-летия пациента или должна быть позже.».  создается ДУЛ и формируется успешный ответ – информация по документу:   1. Person\_id (N, О) – идентификатор пациента; 2. Document\_id (N, О) – идентификатор ДУЛ |
| **Пример** |  |

* 1. Изменение документа

**PUT api/Document**

Редактирование информации по ДУЛ человека.

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, Н) – идентификатор пациента; * Document\_id (N, О) – идентификатор ДУЛ * DocumentType\_id (N, Н) – Тип документа (значение справочника dbo.DocumentType); * Document\_Ser (S, Н) – Серия документа; * Document\_Num (S, Н) – Номер документа; * OrgDep\_id (N, Н) – Организация выдачи документа (значение справочника dbo.OrgDep); * Document\_begDate (D, Н) – Дата выдачи; * KLCountry\_id (N, Н) – Гражданство (значение справочника dbo.KLCountry); * Document\_IsTwoNation (N, Н) – Признак «Двойное гражданство» (0 – Нет, 1 - Да). |
| **Ответ** | Перед тем как изменить данные ДУЛ, если в запросе были хотя бы один из параметров: DocumentType\_id, Document\_Ser, Document\_Num, - производиться проверка на валидность серии документа (Document\_Ser) и номера документа (Document\_Num) (для проверки используются регулярные выражения в таблице dbo.DocumentType: DocumentType\_MaskSer, DocumentType\_MaskNum).  И выполняются следующие проверки:   1. Если по переданному значению Person\_id не найден пациент, то ошибка «Пациент не найден в системе». 2. Если для пациента (Person\_id) не найден документ (Document\_id), то ошибка «Нет данных о документе пациента с указанным Document\_id». 3. Если с учетом передаваемых изменений по параметрам документ с аналогичными параметрами для данного человека найден в системе, обновление документа не происходит, ошибка «Данные документа не прошли проверку на дублирование.». 4. Если в любом передаваемом параметре передается значение, не соответствующее ожидаемому формату, например:  * ожидается числовое значение, а передано текстовое значение; * ожидается параметр, состоящий из 11 символов, а передан параметр, состоящий из большего или меньшего количества символов;   то, ошибка: «Неверный формат поля <Наименования полей через запятую>», обновление документа не производится.   1. Если в передаваемом параметре (DocumentType\_id, OrgDep\_id, KLCountry\_id) передается значение id, не найденное в системе, ошибка: «Значения для <Наименования полей через запятую> не найдены в справочниках системы». Создание документа не происходит. Аналогичный контроль производится для параметров, по которым ожидается значение 0/1 (да/нет) (Document\_IsTwoNation). Если контроль не пройден, то ошибка «Значение поля <названия полей через запятую> не соответствует ожидаемому», создание документа не производится. 2. на дату выдачи в зависимости от типа документа. Если в поле «Тип» выбран любой тип, **кроме**:  * DocumentType\_id = 17 иные документы, * DocumentType\_id = 3 свидетельство о рождении РФ, * DocumentType\_id = 9 паспорт иностранного гражданина, * DocumentType\_id = 19 документ иностранного гражданина, * DocumentType\_id = 22 свидетельство о рождении, выданное не в РФ   то выполняется проверка: переданная в поле Document\_begDate «Дата выдачи» (если поле было передано) дата должна быть больше или равна «Дата рождения» человека Person\_id + 14 лет. Если условие не выполняется, то пользователю выводится сообщение: «Дата выдачи документа Document\_begDate должна соответствовать дате 14-летия пациента или должна быть позже.».  Если найден один человек и пройдены все проверки, то редактируются данные ДУЛ и формируется успешный ответ – пустой ответ с кодом ошибки «0» |
| **Пример** |  |

* 1. Получение ИИН и РПН идентификатор человека

**GET api/PersonBdz**

|  |  |
| --- | --- |
| **Входящие параметры** | * person\_id (N,O) –идентификатор Человека |
| **Ответ** | В ответе возвращается запись со следующими параметрами:   * BDZ\_id (N,О) –идентификатор РПН * PersonInn\_Inn(N,O) –ИИН Человека |
| **Пример** |  |

* 1. Получение записей на прием по МО

**GET api/TimeTableGraf/TimeTableGrafbyMO**

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N, O) - Идентификатор МО * TimeTableGraf\_beg (D, O) - Дата начала диапазона * TimeTableGraf\_end (D, O) - Дата окончания диапазона |
| **Ответ** | Успешный ответ - массив объектов: {   * TimeTableGraf\_id (N, O) – идентификатор бирки. * Person\_id (N, H) – Идентификатор пациента;   } |
| **Пример** |  |

* 1. Получение записей на прием к врачу

**GET api/TimeTableGraf/TimeTableGrafByMedStaffFact**

|  |  |
| --- | --- |
| **Входящие параметры** | * MedStaffFact\_id (N, O) – Место работы врача * TimeTableGraf\_beg (DT, O) - Дата и время начала диапазона * TimeTableGraf\_end (DT, O) - Дата и время окончания диапазона |
| **Ответ** | Успешный ответ – список бирок и пациентов записанных к врачу в указанный диапазон:   * TimeTableGraf\_id (N, O) - Идентификатор бирки * TimeTableGraf\_begTime (DT, О) – Дата и время приема * TimeTableGraf\_factTime (DT, H) – Фактическое время приема. * Person\_id (N, O) – Идентификатор пациента. |
| **Пример** |  |

* 1. Получение прикреплений пациента

**GET api/PersonAttach**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, O) – Идентификатор пациента; * Lpu\_id (N, H) – Идентификатор МО; * LpuRegion\_id (N, H) – Идентификатор участка; * LpuAttachType\_id (N, H) – Идентификатор типа прикрепления. |
| **Ответ** | Успешный ответ:   * PersonSnils\_Snils (N[11], Н) – СНИЛС; * Org\_id (N, Н) – Место работы (значение справочника dbo.Org); * Post\_id (N, Н) – Должность (значение справочника dbo.Post)   Список прикреплений:   * PersonCard\_id (N, Н) – Идентификатор прикрепления; * PersonCard\_begDate (D, Н) – Дата прикрепления; * PersonCard\_endDate (D, Н) – Дата открепления; * PersonCard\_Code (N, H) – Номер амбулаторной карты; * Lpu\_id – Идентификатор МО прикрепления; * LpuRegion\_id (N, H) – Идентификатор участка прикрепления; * LpuAttachType\_id (N, H) – Идентификатор типа прикрепления; * PersonCard\_isAttachAuto (N, H) – Признак автоприкрепления; * CardCloseCause\_id (N, H) – Идентификатор причины закрытия карты. * PersonCard\_AttachAutoDT (DT, H) – Дата и время автоматического прикрепления; * PersonCard\_isAttachCondit (N, H) – Признак условного прикрепления. |
| **Пример** |  |

* 1. Получение списка изменений прикреплений пациентов за период

**Регион: Пенза.**

**GET api/PersonAttachList**

|  |  |
| --- | --- |
| **Входящие параметры** | * bdzID (N, Н) – идентификатор человека в ТФОМС. * LpuAttachType\_id (N,H) – идентификатор типа прикрепления. * PersonCardAttach (N, О) – признак наличия заявления о прикреплении (1 – есть заявление, 0 – нет заявления). * begDate (DT, О) – начало периода. * endDate (DT, О) – окончание периода. * PageNum (N, O) – запрашиваемая страница пакета (страница включает до 10000 записей). |
| **Ответ** | Успешный ответ:  Количество записей в пакете   * ZAP (N, O) – Количество записей в пакете (передается в корне пакета).   Идентификаторы пациента, ФИО, пол, СНИЛС:   * BDZ\_id (N, Н) – идентификатор человека в ТФОМС; * Person\_id (N, О) – идентификатор пациента РМИС; * PersonSurName\_SurName (S[50], О) – Фамилия; * PersonFirName\_FirName (S[50], Н) – Имя; * PersonSecName\_SecName (S[50], Н) – Отчество; * PersonBirthDay\_BirthDay (D, Н) – Дата рождения; * Person\_Sex\_id (N, Н) – Пол (значение справочника dbo.Sex); * PersonSnils\_Snils (N[11],Н) – СНИЛС.   Документ человека (на дату изменения прикрепления):   * Document\_id (N, О) – идентификатор ДУЛ; * DocumentType\_id (N, О) – Тип документа (значение справочника dbo.DocumentType) ; * Document\_Ser (S, Н) – Серия документа; * Document\_Num (S, О) – Номер документа.   Полис человека (на дату изменения прикрепления):   * Polis\_id (N, О) – идентификатор полиса; * PolisType\_id (N, О) – Тип полиса (значение справочника dbo.PolisType) ; * Polis\_Ser (S[10], Н) – Серия полиса; * Polis\_Num (N[15], О) – Номер полиса; * OrgSmoCode (N, О) – Код СМО (значение сущности из dbo.Orgsmo\_f002smocod) ; * Polis\_BegDate (D, О) – Дата выдачи; * Polis\_EndDate (D, Н) – Дата закрытия; * ENP (N[16], О) – ЕНП; * PolisCloseCause\_id (N, О) – идентификатор причины закрытия полиса; * PolisCloseCause\_Name(S, Н) – название причины закрытия полиса.   Данные прикрепления:   * PersonCard\_id (N, Н) – Идентификатор прикрепления; * PersonCard\_begDate (D, Н) – Дата прикрепления; * PersonCard\_endDate (D, Н) – Дата открепления; * CardCloseCause\_id (N, H) – Идентификатор причины закрытия карты; * CardCloseCause\_name (S, H) – Наименование причины закрытия карты.   PersonCard\_isAttachCondit (N, H) – Признак условного прикрепления (2 – условное прикрепление, 1 – заявительное прикрепление). Если PersonCardAttach\_id is not null, то PersonCard\_isAttachCondit =1, иначе PersonCard\_isAttachCondit =2.   * Lpu\_id (N, О) – Идентификатор МО прикрепления; * Lpu\_Code – Код МО прикрепления F003 (dbo.lpu\_f003mcod) ; * LpuRegion\_id (N, H) – Идентификатор участка прикрепления; * LpuRegion\_Name (S, H) – Номер участка прикрепления; * LpuRegion\_tfoms (S, H) – Номер участка прикрепления ТФОМС; * LpuSection\_id (N, H) – идентификатор отделения участка прикрепления; * LpuBuilding\_id (N, H) – идентификатор подразделения участка прикрепления; * LpuSection\_CodeTFOMS (S, H) – код отделения участка прикрепления ТФОМС (берется из атрибутов связанного отделения) ; * LpuBuilding\_CodeTFOMS (S, H) – код подразделения участка прикрепления ТФОМС (берется из атрибутов связанного отделения); * Doc\_Snils (N[11],Н) – СНИЛС основного врача участка; * Doc\_Type (N[11],Н) – тип должности врача (1 – Врач, 2 – Ср. мед. персонал); * LpuRegionF\_id (N, H) – Идентификатор участка прикрепления (ФАП); * LpuRegionF\_Name (S, H) – Номер участка прикрепления (ФАП); * LpuSectionF\_id (N, H) – идентификатор отделения участка прикрепления (ФАП); * LpuBuildingF\_id (N, H) – идентификатор подразделения участка прикрепления (ФАП); * LpuSectionF\_CodeTFOMS (S, H) – код отделения участка прикрепления ТФОМС (берется из атрибутов связанного отделения) ФАП; * LpuBuildingF\_CodeTFOMS (S, H) – код подразделения участка прикрепления ТФОМС (берется из атрибутов связанного отделения) ФАП; * DocF\_Snils (N, Н) – СНИЛС основного врача участка (ФАП); * DocF\_Type (N, Н) – тип должности врача (ФАП) (1 – Врач, 2 – Ср. мед. персонал). |
| **Пример** |  |

В формируемом ответе формата JSON тег количества записей (ZAP) в пакете размещается первым, на том же уровне, что и тег с данными (DATA).

В пакете передаются не более 10000 записей.

Выборка для страницы осуществляется в зависимости от значения входного параметра «Заправшиваемая страница пакета» (PageNum).

Если номер страницы «1», то должны быть переданы с 1 по 10000 записи выборки (или меньше, если количество записей по запросу менее 10000).

Если номер страницы «2», то с 10001 по 20000 (или меньше, если количество записей по запросу менее 20000), и т.д.

* 1. Получение согласия на обработку персональных данных.

**GET api/PersonLpuInfo\_IsAgree**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, O) – Идентификатор пациента; * Lpu\_id (N, O) – Идентификатор МО; |
| **Ответ** | Успешный ответ:   * PersonLpuInfo\_IsAgree (N, O) – согласие на обработку персональных данных. |
| **Пример** |  |

* 1. Получение данных по сигнальной информации пациента

**GET api/PersonSignalInfo**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, O) – Идентификатор пациента |
| **Ответ** | Успешный ответ:   * PersonMedHistory\_id (N, H) – Идентификатор анамнеза жизни пациента; * PersonMedHistory\_Descr (T, H) – Текст анамнеза жизни (в редакторе); * PersonMedHistory\_Text (T, H) – Текст анамнеза жизни; * PersonMedHistory\_setDT (D, H) – Дата установки анамнеза; * PersonBloodGroup\_id (N, H) – Идентификатор группы крови и резус-фактора; * BloodGroupType\_id (N, H) – Идентификатор группы крови; * BloodGroupType\_Code (N, H) – Код группы крови; * BloodGroupType\_Name (T[100], H) – Наименование группы крови; * RhFactorType\_id (N, H) – Идентификатор резус-фактора; * RhFactorType\_Code (N, H) – Код резус-фактора; * RhFactorType\_Name (T[100], H) – Наименование резус-фактора; * PersonBloodGroup\_setDT (D, H) – Дата определения группы крови;   Список аллергических реакций: массив PersonAllergicReactionList{   * PersonAllergicReaction\_id (N, H) – Идентификатор аллергической реакции; * AllergicReactionLevel\_id (N, H) – Идентификатор характера аллергической реакции * AllergicReactionLevel\_Name (T, H) – Наименование характера аллергической реакции * AllergicReactionType\_id (N, H) – Идентификатор типа аллергической реакции * AllergicReactionType\_Name (T, H) – Наименование типа аллергической реакции * DrugMnn\_id (N, H) – Лекарственный препарат-аллерген * PersonAllergicReaction\_Kind (T[255], H) – Вид аллергена * PersonAllergicReaction\_setDT (D, H) – Дата возникновения аллергической реакции   }  Список экспертных анамнезов и льгот: массив PrivilegeTypeList{   * PrivilegeType\_id (N, H) – Идентификатор анамнезов и льгот; * PrivilegeType\_Name (T[150], H) – Наименование анамнезов и льгот; * PrivilegeType\_Code (N, H) – Код анамнезов и льгот; * PersonPrivilege\_begDate (D, H) – Дата начала анамнеза и льготы; * PersonPrivilege\_endDate (D, H) – Дата окончания и льготы;   }  Список диспансерных наблюдений пациента: массив PersonDispList{   * Diag\_id (N, H) – Идентификатор диагноза; * PersonDisp\_begDate (D, H) – Дата постановки на учет; * PersonDisp\_endDate (D, H) – Дата снятия с учета; * Diag\_Code (T[10], H) – Код диагноза в МКБ-10; * DispOutType\_id (N, H) – Идентификатор причины снятия с учета; * DispOutType\_Name (T, H) – Причина снятия с учета; * LpuSection\_id (N, O) – Идентификатор отделения; * LpuSectionProfile\_id (N, O) – Идентификатор профиля отделения; * MedPersonal\_id (N, O) – Идентификатор медицинского работника;   }  Список диспансеризаций и медицинских осмотров: массив EvnPLDispList{   * DispClass\_id (N, O) – Идентификатор вида диспансеризации; * HealthKInd\_id (N, H) – Идентификатор группы здоровья; * Lpu\_id (N, O) – Идентификатор МО проведения осмотра/диспансеризации; * EvnPLDisp\_consDT (D, H) – Дата формирования информационного добровольного согласия * EvnPLDisp\_setDT (D, H) – Дата осмотра услуги педиатра или терапевта (по времени выполнения услуги Evn\_setDT); * EvnPlDisp\_IsEndStage (N, H) – Показатель законченности случая;   Список диагнозов, впервые установленных (DeseaseDispType\_id=2): подмассив EvnPLDispDiagList{   * Diag\_id (N, H) – Идентификатор диагноза; * Diag\_Code (T[10], H) – Код диагноза в МКБ-10;   }  }  Список уточненных диагнозов по EvnDiagSpec, EvnSection, EvnDiagPS, EvnVizitPL, EvnDiagPLSop: массив PersonDiagOsnList{   * Diag\_id (N, H) – Идентификатор диагноза; * Diag\_Code (T[10], H) – Код диагноза в МКБ-10; * Lpu\_id (Т, H) – Идентификатор МО; * LpuSectionProfile\_name (Т, H) – Наименование профиля отделения; * EvnDiag\_setDT (D, H) – Дата установки диагноза; * MedStaffFact\_id (N, H) – Идентификатор места работы врача, установившего диагноз;   }  Список оперативных вмешательств: массив EvnUslugaOperList{   * EvnUslugaOper\_setDT (DT, H) – Дата и время начала выполнения услуги (по дате и времени начала выполнения услуги Evn\_setDT); * EvnUslugaOper\_disDT (DT, H) – Дата и время окончания выполнения услуги (по времени окончания выполнения услуги Evn\_disDT); * UslugaComplex\_Name (N, H) – наименование оперативной услуги; * UslugaComplex\_Code (N, H) – Код оперативной услуги; * Lpu\_id (N, O) – Идентификатор МО выполнения оперативной услуги;   }  Список отмененных направлений: массив EvnDirectionFailList{   * EvnDirection\_id (N, H) – Идентификатор отмененного направления; * MedStaffFact\_id (N, H) – Идентификатор места работы направившего врача; * EvnDirection\_failDT (D, H) – Дата отмены направления; * EvnDirection\_setDate (D, Н) – Дата направления; * EvnStatusCause\_id (N, H) – Идентификатор причины отклонения; * pmUser\_Name (T, H) – ФИО пользователя, отменившего направление (определяем по пользователю pmUserfailID);   }  Антропометрические данные:   * PersonHeight\_id (N, H) – Идентификатор роста человека; * PersonHeight\_setDT (D, H) – Дата измерения роста; * PersonHeight\_Height (N, H) – Рост человека (в см); * PersonHeight\_isAbnorm (N, H) – Отклонение роста от нормы; * HeightAbnormType\_id (N, H) – Идентификатор типа отклонения роста от нормы; * HeightMeasureType\_id (N, H) – Идентификатор вида замера роста; * PersonWeight\_id (N, H) – Идентификатор веса человека; * PersonWeight\_setDT (D, H) – Дата измерения веса; * PersonWeight\_Weight (N, H) – Вес человека; * PersonWeight\_isAbnorm (N, H) – Отклонение веса от нормы; * WeightAbnormType\_id (N, H) – Идентификатор типа отклонения веса от нормы; * WeightMeasureType\_id (N, H) – Идентификатор вида замера веса; * Okei\_id (N, H) – Идентификатор единицы измерения веса.   Список открытых ЛВН: массив OpenEvnStickList{   * EvnStick\_SetDate (D, H) – Дата выдачи ЛВН; * StickWorkType\_id (N, H) – Тип занятости (значение справочника dbo.StickWorkType); * EvnStick\_Num (N, H) – Номер ЛВН; * EvnStick\_Ser (N, H) – Серия ЛВН; * StickOrder\_id (N, H) – Порядок выдачи ЛВН (значение справочника dbo.StickOrder).   } |

* 1. Получение данных по амбулаторной карте по пациенту и МО

**GET api/PersonAmbulatCard**

Получение информации по амбулатроной карте по пациенту и МО.

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, O) – Идентификатор пациента; * Lpu\_id (N, H) – Идентификатор МО. |
| **Ответ** | Успешный ответ список амбулаторных карт с данными:   * Lpu\_id (N, H) – Идентификатор МО прикрепления; * PersonAmbulatCard\_id (N, О) – Идентификатор амбулаторной карты; * PersonAmbulatCard\_Num (N, О) – Номер амбулаторной карты; * PersonAmbulatCard\_begDate (N, О) – Дата открытия карты; * PersonAmbulatCard\_endDate (N, Н) – Дата закрытия карты; * LpuAttachType\_id (N, O) – Идентификатор типа прикрепления к МО. |

1. Описание методов работы с логами федеральных сервисов
2. 1. Получение лога по человеку и МО

**GET api/Logger/Person**

Идентификатор МО получаем из сессии пользователя.

Идентификатор человека получаем в результате вызова метода GET api/PersonList

Данные берем из таблицы iemc.PersonHistory

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N, O) – Идентификатор МО. * Person\_id (N, O) – Идентификатор человека. * LogPeriod\_beg (DT, H) - Дата и время начала периода * LogPeriod\_end (DT, H) - Дата и время окончания периода |
| **Ответ** | Успешный ответ: обе даты в ответе должны попадать в запрошенный период времени.   * PersonHistory\_id (N, O) – Идентификатор лога * PersonHistory\_insDT (DT, O) – Дата и время ввода. * PersonHistory\_getDT (DT, O) – Дата и время запроса. |
| **Пример** |  |

* 1. Получение лога по паспорту медицинского изделия

**GET api/Logger/PMI**

Идентификатор МО получаем из сессии пользователя. Данные берем из таблицы passport.LoggerPMI

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N, O) – Идентификатор МО. * LogPeriod\_beg (DT, H) - Дата и время начала периода * LogPeriod\_end (DT, H) - Дата и время окончания периода |
| **Ответ** | Успешный ответ: обе даты в ответе должны попадать в запрошенный период времени.   * LoggerPMI\_id (N, O) – Идентификатор лога. * LoggerPMI\_InventNumber (T[64], O) – Инвентарный номер изделия. * LoggerPMI\_isStatusOK (N, O) – Статус записи. Значение типа флаг. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * LoggerPMI\_Message (T[4000], O) – Описание. * LoggerPMI\_setDT (DT, O) – Дата и время запуска обработки партии * LoggerPMI\_insDT(DT, O) – Дата и время создания. * LoggerPMI\_updDT(DT, O) – Дата и время изменения. |
| **Пример** |  |

* 1. Получение лога по паспорту МО

**GET api/Logger/PMU**

Идентификатор МО получаем из сессии пользователя. Данные берем из таблицы passport.LoggerPMU

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N, O) – Идентификатор МО. * LogPeriod\_beg (DT, H) - Дата и время начала периода * LogPeriod\_end (DT, H) - Дата и время окончания периода |
| **Ответ** | Успешный ответ: обе даты в ответе должны попадать в запрошенный период времени.   * LoggerPMU\_id (N, O) – Идентификатор лога. * LoggerPMU\_isStatusOK (N, O) – Статус записи. Значение типа флаг. Возможные значения: 0 и 1, - где 0 – нет, 1 – да * LoggerPMU\_Message (T[4000], O) – Описание. * LoggerPMU\_setDT (DT, O) – Дата и время запуска обработки партии * LoggerPMU\_insDT(DT, O) – Дата и время ввода. * LoggerPMU\_updDT(DT, O) – Дата и время сохранения. |
| **Пример** |  |

* 1. Получение лога отправки событий (КВС и ТАП)

**GET api/Logger/EvnPL**

Отправленное в ИЭМК создание какого-то события в рамках ТАП или КВС сначала проверяется на ФЛК. Если не проходит, то возвращает ошибку. Если проходит, то возвращает идентификатор события.

Данные берем из таблицы iemc.IntegrationServiceEventLog

В таблице iemc.IntegrationServiceEvent ищем в поле requestParams полученные идентификаторы событий. Входящий параметр в метод находим по полю IntegrationServiceEvent.Id.

|  |  |
| --- | --- |
| **Входящие параметры** | * eventId (N, O) – Идентификатор события. * LogPeriod\_beg (DT, H) - Дата и время начала периода * LogPeriod\_end (DT, H) - Дата и время окончания периода |
| **Ответ** | Успешный ответ: дата события в ответе должна попадать в запрошенный период времени.   * IntegrationServiceEventLog.bodyType (T[256], O) – Тип ответа. * IntegrationServiceEventLog.description (T[-1], O) – Описание. * IntegrationServiceEventLog.eventDate (DT, O) – Дата и время события. * IntegrationServiceEventLog.id (N, O) – Идентификатор лога. * IntegrationServiceEventLog.level (T[255], O) – уровень ошибки (ERROR или INFO). * IntegrationServiceEventLog.title (T[255], O) – Заголовок лога. |
| **Пример** |  |

1. Описание методов работы со спецификами
   1. Получение данных по специфике новорожденного

**GET api/PersonNewborn**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, Н) – Ид человека; * EvnSection\_mid (N, Н) – Ид движения матери, из которого добавлена специфика |
| **Ответ** | Проверка: Хотя бы один из параметров Person\_id, EvnSection\_mid должен быть заполнен. Иначе ошибка.  Успешный ответ:   * PersonNewborn\_id – Ид специфики; * Person\_id – Ид человека; * BirthSvid\_id – Ид свидетельства о рождении; * EvnPS\_id – Ид КВС; * BirthSpecStac\_id – Ид специфики по родам; * EvnSection\_mid – Ид движения матери, из которого добавлена специфика; * FeedingType\_id – Ид способа вскармливания; * ChildTermType\_id – Ид признака доношенности; * PersonNewborn\_IsAidsMother – Признак «ВИЧ-инфекция у матери»; * PersonNewborn\_CountChild – Который по счету; * ChildPositionType\_id – Ид типа предлежания; * PersonNewborn\_IsRejection – Признак «Отказ от ребенка»; * PersonNewborn\_IsHighRisk – Признак «Высокий риск»; * NewbornWardType\_id – Ид типа палаты для новорожденных; * PersonNewborn\_IsBleeding – Признак «Наличие кровотечения»; * PersonNewborn\_IsNeonatal – Признак «Неонатальный скрининг»; * PersonNewborn\_IsAudio – Признак «Аудиологический скрининг»; * PersonNewborn\_IsBCG – Признак «БЦЖ»; * PersonNewborn\_BCGDate – Дата БЦЖ; * PersonNewborn\_BCGSer – БЦЖ серия; * PersonNewborn\_BCGNum – БЦЖ номер; * PersonNewborn\_IsHepatit – Признак «Гепатит B» * PersonNewborn\_HepatitDate – Гепатит B дата; * PersonNewborn\_HepatitSer – Гепатит B серия; * PersonNewborn\_HepatitNum – Гепатит B номер; * PersonNewborn\_Head – Окружность головы, см; * PersonNewborn\_Breast – Окружность груди, см; * PersonNewborn\_Height – Рост при рождении, см; * PersonNewborn\_Weight – Вес при рождении, г; * PersonNewborn\_IsBreath – Признак «Дыхание»; * PersonNewborn\_IsHeart – Признак «Сердцебиение»; * PersonNewborn\_IsPulsation – Признак «Пульсация пуповины»; * PersonNewborn\_IsMuscle – Признак «Произвольное сокращение мускулатуры»; * PersonNewborn\_BloodBili – Общий билирубин, Ммоль/л; * PersonNewborn\_BloodHemoglo – Гемоглобин, г/л; * PersonNewborn\_BloodEryth – Эритроциты, 1012/л; * PersonNewborn\_BloodHemato – Гематокрит, % |
| **Пример** |  |

* 1. Получение данных по специфике родов

**GET api/BirthSpecStac**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnSection\_id (N, О) – Ид движения матери, из которого добавлена специфика; * PregnancyResult\_id (N, Н) – Тип результата беременности. |
| **Ответ** | Успешный ответ:   * BirthSpecStac\_id – Ид специфики; * EvnSection\_id – Ид движения матери, из которого добавлена специфика; * BirthSpecStac\_CountPregnancy – Которая по счету беременность; * BirthSpecStac\_CountBirth – Которые по счету роды; * BirthSpecStac\_CountChild – Количество плодов; * BirthSpecStac\_CountChildAlive – Количество плодов живорожденных; * BirthResult\_id – Характер родов; * BirthPlace\_id – Место родов; * BirthSpecStac\_OutcomPeriod – Срок исхода (нед.); * BirthSpecStac\_OutcomDT – Дата и время исхода; * BirthSpec\_id – Особенности родов; * BirthSpecStac\_IsHIVtest – Признак «Обследована на ВИЧ»; * BirthSpecStac\_IsHIV – Признак «Наличие ВИЧ-инфекции»; * AbortType\_id – Тип аборта; * BirthSpecStac\_IsMedicalAbort – Признак «Медикаментозный аборт»; * BirthSpecStac\_BloodLoss – Кровопотери (мл); * PregnancySpec\_id – Ид специфики по беременности; * Evn\_id – Ид события; * Lpu\_id – МО исхода; * PregnancyResult\_id – Тип результата беременности; * BirthCharactType\_id – Характер родов; * AbortLpuPlaceType\_id – Место аборта; * BirthSpecStac\_IsRWtest – Признак «Обследована на сифилис»; * BirthSpecStac\_IsRW – Признак «Наличие сифилиса»; * BirthSpecStac\_IsHBtest – Признак «Обследована на гепатит B»; * BirthSpecStac\_IsHB – Признак «Наличие гепатита B»; * BirthSpecStac\_IsHCtest – Признак «Обследована на гепатит C»; * BirthSpecStac\_IsHC – Признак «Наличие гепатита C»; * AbortLawType\_id – Вид аборта; * AbortMethod\_id – Метод аборта; * BirthSpecStac\_InjectVMS – Введено ВМС; * BirthSpecStac\_Info – Сведения; * BirthSpecStac\_SurgeryVolume – Объем оперативное вмешательства при вниматочной беременности; * AbortIndicat\_id – Показание для аборта; * BirthSpecStac\_IsContrac – Признак «Послеродовая контрацепция»; * BirthSpecStac\_ContracDesc – Сведения о послеродовой контрацепции; * PersonRegister\_id – Ид регистра |
| **Пример** |  |

* 1. Получение данных по специфике психиатрии, наркологии

**GET api/MorbusCrazyPerson**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, Н) – Ид человека; * Evn\_pid (N, Н) – Ид события-родителя (связь в таблице dbo.MorbusBase); * MorbusType (N, Н) – Тип специфики (1 – Психиатрия, 2 – Наркология). |
| **Ответ** | Проверка: хотя бы один из параметров Person\_id, Evn\_pid должен быть заполнен. Иначе ошибка.  Если MorbusType = «1. Психиатрия», то выводятся данные только по спецификам психиатрии (диагнозы из групп F, за исключением группы F10.0 – F19.9).  Если MorbusType = «2. Наркология», то выводятся данные только по спецификам наркологии (диагнозы из группы F10.0 – F19.9).  Если MorbusType НЕ зполнено, то выводятся данные и по спецификам психиатрии, и по спецификам наркологии.  Успешный ответ:   * Person\_id – Ид человека; * Evn\_pid – Ид события-родителя (связь в таблице dbo.MorbusBase); * MorbusBase\_id – Ид заболевания; * MorbusType – Тип специфики (1 – Психиатрия, 2 – Наркология);   -- Данные из регистра по психиатрии/наркологии (MorbusType\_id=4):   * PersonRegister\_id – Ид регистра; * Diag\_id – Диагноз регистра; * PersonRegister\_setDT – Дата включения в регистр;   --- Данные из специфики психиатрии/наркологии:   * MorbusCrazyPerson\_id – Ид специфики; * Person\_id – Ид человека; * MorbusCrazyPerson\_IsWowInvalid – Признак «Инвалид ВОВ»; * MorbusCrazyPerson\_IsWowMember – Признак «Участник ВОВ»; * CrazyEducationType\_id – Образование; * MorbusCrazyPerson\_CompleteClassCount – Число законченных классов среднеобразовательного учреждения; * MorbusCrazyPerson\_IsEducation – Признак «Учится»; * CrazySourceLivelihoodType\_id – Источник средств существования; * CrazyResideType\_id – Проживает; * CrazyResideConditionsType\_id – Условия проживания; * MorbusCrazyPerson\_IsConvictionBeforePsych – судимости до обращения к психиатру (наркологу).   --- Данные по диагнозу (используется запись с максимальной MorbusCrazyDiag\_setDT):   * MorbusCrazyDiag\_id – Ид дианоза; * MorbusCrazy\_id – Заболевание; * MorbusCrazyDiag\_setDT – Дата установления (пересмотра); * CrazyDiag\_id – Диагноз; * Diag\_sid – Сопутствующий дианоз.   --- Данные по инвалидности (используется запись с максимальной MorbusCrazyPersonInvalid\_setDT):   * MorbusCrazyPersonInvalid\_id – Ид инвалидности; * MorbusCrazyPersonInvalid\_setDT – Дата установления (пересмотра); * InvalidGroupType\_id – Группа инвалидности; * MorbusCrazyPersonInvalid\_reExamDT – Срок очередного переосвидетельствования; * MorbusCrazyPersonInvalid\_Article – Статья УК; * CrazyWorkPlaceType\_id – Место работы. |
| **Пример** |  |

* 1. Получение данных по употребляемым психоактивным веществам

**GET api/MorbusCrazyDrug**

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusBase\_id (N, О) – Ид заболевания. |
| **Ответ** | Успешный ответ – список употребляемых психоактивных веществ с параметрами:   * MorbusCrazyDrug\_id – Ид приема; * MorbusCrazyBase\_id – Ид заболевания; * MorbusCrazyDrug\_Name – Наименование психоактивного вещества; * CrazyDrugType\_id – Вид вещества; * CrazyDrugReceptType\_id – Тип приема. |
| **Пример** |  |

* 1. Создание извещения об инфекционном заболевании (ВИЧ)

POST api/EvnInfectNotify

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_pid (N,O) – ссылка на случай * EvnInfectNotify\_DiseaseDate (T, О) – Дата заболевания * EvnInfectNotify\_FirstMeasures (T, Н) – Проведенные первичные противоэпидемические мероприятия и дополнительные сведения * EvnInfectNotify\_FirstSESDT (DT, Н) – Дата и час первичной сигнализации в СЭС * EvnInfectNotify\_FirstTreatDate (D, О) – Дата первичного обращения (выявления) * EvnInfectNotify\_IsLabDiag (N, О) – Диагноз подтвержден лабораторно (=2, если «1.Да»; =1, если «2.Нет»). * EvnInfectNotify\_NextVizitDate (D, Н) – Дата последующего посещения детского учреждения, школы * Lpu\_id (N,O) – Место госпитализации * EvnInfectNotify\_PoisonDescr (T, Н) – Где произошло отравление, чем * EvnInfectNotify\_ReceiverMessage (T, Н) – Кто принял сообщение * EvnInfectNotify\_SetDiagDate (D, О) – Дата установления диагноза * MedPersonal\_id (N, Н) – Врач, создавший извещение |
| **Ответ** | **Успешный ответ идентификатор записи**   * EvnInfectNotify\_id (N, O) – Идентификатор извещения |
| **Пример** |  |

* 1. Получение извещения об инфекционном заболевании (ВИЧ)

GET api/EvnInfectNotify

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnInfectNotify\_id (N, Н) – Идентификатор извещения * Person\_id (N, Н) – ссылка на человека в Person |
| **Ответ** | Хотя бы один из входящих параметров должен быть заполнен, иначе ошибка (не передан ни один из параметров).   * Person\_id (N, O) – ссылка на человека в Person * EvnInfectNotify\_DiseaseDate (T, О) – Дата заболевания * EvnInfectNotify\_FirstMeasures (T, Н) – Проведенные первичные противоэпидемические мероприятия и дополнительные сведения * EvnInfectNotify\_FirstSESDT (DT, Н) – Дата и час первичной сигнализации в СЭС * EvnInfectNotify\_FirstTreatDate (D, О) – Дата первичного обращения (выявления) * EvnInfectNotify\_IsLabDiag (N, О) – Диагноз подтвержден лабораторно (=2, если «1.Да»; =1, если «2.Нет»). * EvnInfectNotify\_NextVizitDate (D, Н) – Дата последующего посещения детского учреждения, школы * Lpu\_id (N,O) – Место госпитализации * EvnInfectNotify\_PoisonDescr (T, Н) – Где произошло отравление, чем * EvnInfectNotify\_ReceiverMessage (T, Н) – Кто принял сообщение * EvnInfectNotify\_SetDiagDate (D, О) – Дата установления диагноза * EvnSection\_id (N, Н) – Ссылка на учетный документ (движение, посещение) * Diag\_id (N, O) – Идентификатор диагноза * MedPersonal\_id (N, Н) – Врач, создавший извещение |
| **Пример** |  |

* 1. Создание оперативного донесения о лице, в крови которого при исследовании в реакции иммуноблота выявлены антитела к ВИЧ

POST api/EvnNotifyHIV

|  |  |
| --- | --- |
| **Входящие параметры** | * Evn\_pid (N,O) – ссылка на случай * HIVContingentTypeP\_id (N,H) – Гражданство (значения 100 или 200 из справочника dbo.HIVContingentType) * Контингент: массив HIVContingentType\_Name\_list{ * HIVContingentType\_id (N,МH) – идентификатор контингента (справочник dbo.HIVContingentType)} * MorbusHIVLab\_BlotDT (D, H) – Дата иммунного блоттинга * MorbusHIVLab\_BlotNum (T, H) – № серии * MorbusHIVLab\_BlotResult (T, H) – Выявленные белки и гликопротеиды * MorbusHIVLab\_TestSystem (T, H) – Тип тест–системы * LabAssessmentResult\_iid (N, У) – Результат реакции иммуноблота (справочник dbo.LabAssessmentResult) (параметр обязательный, если указан MorbusHIVLab\_BlotDT) * Lpu\_id (N,H) – Учреждение, первично выявившее положительный результат в ИФА * MorbusHIVLab\_IFADT (D, H) – Дата иммуноферментного анализа * MorbusHIVLab\_IFAResult (T, H) – Результат иммуноферментного анализа * MorbusHIV\_confirmDate (D, Н) – Дата подтверждения диагноза * MorbusHIV\_EpidemCode (T, Н) – Эпидемиологический код * MedPersonal\_id (N, О) – Врач, создавший извещение * EvnNotifyHIV\_setDT (D, О) * PersonRegisterType\_id (N, Н) – тип регистра (справочник dbo.PersonRegisterType)   --При добавлении в регистр обязательно заполняются поля:   * Lpu\_iid (N, У) – ЛПУ, добавившее человека в регистр * PersonRegister\_setDate (D, У) – Дата включения в регистр * MedPersonal\_iid (N, У) – врач, добавивший человека в регистр * PersonRegisterType\_id (N, У) – тип регистра (справочник dbo.PersonRegisterType) |
| **Ответ** | * EvnNotifyHIV\_id (N, O) – идентификатор донесения * PersonRegister\_id (N, У) – идентификатор записи регистра (при автоматическом добавлении в регистр). * Morbus\_id (N, O) – идентификатор заболевания |
| **Пример** |  |

* 1. Получение оперативного донесения о лице, в крови которого при исследовании в реакции иммуноблота выявлены антитела к ВИЧ

GET api/EvnNotifyHIV

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnNotifyHIV\_id (N, Н) – идентификатор донесения * Person\_id (N, Н) – Идентификатор человека |
| **Ответ** | Если не передан ни один параметр, то ошибка (не передан ни один из параметров).  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список с параметрами:   * EvnNotifyHIV\_id (N, О) – идентификатор донесения * Person\_id (N, О) – Идентификатор человека * MorbusHIVLab\_id (N, Н) – Идентификатор лабораторной диагностики * MorbusHIV\_id (N, O) – Заболевание ВИЧ * Diag\_id (N, O) – Идентификатор диагноза * HIVContingentTypeP\_id (N,H) – Гражданство (значения 100 или 200 из справочника dbo.HIVContingentType) * Контингент: массив HIVContingentType\_Name\_list{ * HIVContingentType\_id (N,МH) – идентификатор контингента (справочник dbo.HIVContingentType)} * MorbusHIVLab\_BlotDT (D, H) – Дата иммунного блоттинга * MorbusHIVLab\_BlotNum (T, H) – № серии * MorbusHIVLab\_BlotResult (T, H) – Выявленные белки и гликопротеиды * MorbusHIVLab\_TestSystem (T, H) – Тип тест–системы * LabAssessmentResult\_iid (N, У) – Результат реакции иммуноблота (справочник dbo.LabAssessmentResult) (параметр обязательный, если указан MorbusHIVLab\_BlotDT) * Lpu\_id (N,H) – Учреждение, первично выявившее положительный результат в ИФА * MorbusHIVLab\_IFADT (D, H) – Дата иммуноферментного анализа * MorbusHIVLab\_IFAResult (T, H) – Результат иммуноферментного анализа * MorbusHIV\_confirmDate (D, Н) – Дата подтверждения диагноза * MorbusHIV\_EpidemCode (T, Н) – Эпидемиологический код * MedPersonal\_id (N, О) – Врач, создавший извещение * EvnNotifyHIV\_setDT (D, О) * PersonRegisterType\_id (N, Н) – тип регистра (справочник dbo.PersonRegisterType)   --Обязательные параметры при отказе включения в регистр:   * Lpu\_niid (N, У) – Лпу, не включившее в регистр * MedPersonal\_niid (N, У) – Врач, не включивший в регистр * PersonRegisterFailIncludeCause\_id (N, У) – Причина невключения в регистр (справочник dbo.PersonRegisterFailIncludeCause) |
| **Пример** |  |

* 1. Создание записи регистра по онкологии, больных туберкулезом и ВИЧ-инфицированых

POST api/PersonRegister

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, O) – Идентификатор человека * Diag\_id (N, O) – Идентификатор диагноза * Lpu\_iid (N, O) – ЛПУ, добавившее человека в регистр * MedPersonal\_iid (N, O) – врач, добавивший человека в регистр * PersonRegister\_setDate (D, O) – Дата включения в регистр * PersonRegisterType\_id (N, O) – Тип регистра (значение из справочника dbo.PersonRegisterType) * MorbusType\_id (N, Н) – тип заболевания (значение из справочника dbo.MorbusType) * Morbus\_confirmDate (D, Н) – Дата подтверждения диагноза (если диагноз из диапазона B20-B24) * Morbus\_EpidemCode (T, Н) – Эпидемиологический код (если диагноз из диапазона B20-B24)   --Если включение производится по извещению, обязательно передается идентификатор одного из извещений, по которому пациент включается в регистр:   * EvnNotifyHIV\_id (N, У) – идентификатор оперативного донесения о лице, в крови которого выявлены антитела к ВИЧ (если PersonRegisterType\_id =9) * EvnNotifyTub\_id (N, У) – идентификатор извещения о туберкулезе (если PersonRegisterType\_id =7) * EvnOnkoNotify\_id (N, У) – идентификатор извещения об онкобольном (если PersonRegisterType\_id =3) |
| **Ответ** | **Успешный ответ идентификатор записи**   * PersonRegister\_id (N, O) – идентификатор записи регистра. * MorbusHIV\_id (N, У) – Идентификатор специфики по ВИЧ (обязательный, если PersonRegisterType\_id=9) * MorbusTub\_id (N, У) – идентификатор специфики по туберкулезу (обязательный, если PersonRegisterType\_id=7) * MorbusOnko\_id (N, У) – идентификатор специфики по онкологии (обязательный, если PersonRegisterType\_id=3)   Контроль:  Создание записи регистра недоступно, если пациент уже имеет запись с данным типом регистра, ошибка «Для выбранного пациента уже существует запись регистра» |
| **Пример** |  |

* 1. Получение записи регистра по онкологии, больных туберкулезом и ВИЧ-инфицированых

GET api/PersonRegister

|  |  |
| --- | --- |
| **Входящие параметры** | * PersonRegister\_id (N, Н) – идентификатор записи регистра * Person\_id (N, Н) – Идентификатор человека * PersonRegisterType\_id (N, O) – Тип регистра (значение из справочника dbo. PersonRegisterType) |
| **Ответ** | Если не передан ни один параметр, то ошибка (не передан ни один из параметров).  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список с параметрами:   * PersonRegister\_id (N, О) – идентификатор записи регистра * Person\_id (N, O) – Идентификатор человека * Diag\_id (N, O) – Идентификатор диагноза * Lpu\_iid (N, O) – ЛПУ, добавившее человека в регистр * MedPersonal\_iid (N, O) – врач, добавивший человека в регистр * PersonRegister\_setDate (D, O) – Дата включения в регистр * PersonRegisterType\_id (N, O) – Тип регистра (значение из справочника dbo. PersonRegisterType) * MorbusType\_id (N, Н) – тип заболевания (значение из справочника dbo. MorbusType) * Morbus\_confirmDate (D, Н) – Дата подтверждения диагноза (если диагноз из диапазона B20-B24) * Morbus\_EpidemCode (T, Н) – Эпидемиологический код (если диагноз из диапазона B20-B24)   --Если включение производилось по извещению, обязательно передается идентификатор извещения:   * EvnNotifyHIV\_id (N, У) – идентификатор оперативного донесения о лице, в крови которого выявлены антитела к ВИЧ * EvnNotifyTub\_id (N, У) – идентификатор извещения о туберкулезе (для PersonRegisterType\_id =7) * EvnOnkoNotify\_id (N, У) – идентификатор извещения об онкобольном (для PersonRegisterType\_id =3)   --При исключении из регистра обязательные параметры:   * Lpu\_did (N, У) – ЛПУ, исключившее человека из регистра * MedPersonal\_did (N, У) – врач, исключивший человека из регистра * PersonRegister\_disDate (D, У) – Дата исключения из регистра, * PersonRegisterOutCause\_id (D, У) – Причина исключения из регистра (справочник dbo.PersonRegisterOutCause) * PersonDeathCause\_id (N, У) – Причина смерти, обязательно, если PersonRegisterOutCause\_id=1 (справочник dbo.PersonDeathCause) |
| **Пример** |  |

* 1. Изменение записи регистра по онкологии, больных туберкулезом и ВИЧ-инфицированых

PUT api/PersonRegister

|  |  |
| --- | --- |
| **Входящие параметры** | * PersonRegister\_id (N, O) – Идентификатор записи регистра * Person\_id (N, Н) – Идентификатор человека * Diag\_id (N, Н) – Идентификатор диагноза * Lpu\_iid (N, Н) – ЛПУ, добавившее человека в регистр * MedPersonal\_iid (N, Н) – врач, Добавивший человека в регистр * PersonRegister\_setDate (D, Н) – Дата включения в регистр * PersonRegisterType\_id (N, Н) – Тип регистра (значение из справочника dbo. PersonRegisterType) * MorbusType\_id – тип заболевания (значение из справочника dbo. MorbusType) * Morbus\_confirmDate (D, Н) – Дата подтверждения диагноза * Morbus\_EpidemCode (T, Н) – Эпидемиологический код (если диагноз из диапазона B20-B24) * PersonRegister\_disDate (D, У) – Дата исключения из регистра (обязательный при исключении из регистра) * Lpu\_did (N, У) – ЛПУ, исключившее человека из регистра (обязательный при исключении из регистра) * MedPersonal\_did (N, У) – врач, исключивший человека из регистра (обязательный при исключении из регистра) * PersonRegisterOutCause\_id (D, У) – Причина исключения из регистра (справочник dbo.PersonRegisterOutCause) (обязательный при исключении из регистра) * PersonDeathCause\_id (N, У) – Причина смерти (справочник dbo.PersonDeathCause) |
| **Ответ** | Успешный ответ с ошибкой «0».  Проводятся проверки:   * Параметры Lpu\_did, MedPersonal\_did, PersonRegister\_disDate,   PersonRegisterOutCause передаются или не передаются одновременно, если какие–то из данных параметров не переданы, то ошибка (отсутствует обязательный параметр).   * Параметр PersonDeathCause\_id обязателен, если передана причина исключения из регистра со значением смерть (PersonRegisterOutCause\_id=1), иначе ошибка (отсутствует обязательный параметр). * Дата включения в регистр (PersonRegister\_setDate) не может быть меньше или равна дате включения в регистр (PersonRegister\_disDate), иначе ошибка «Дата исключения из регистра не может быть меньше или равна дате включения в регистр». |
| **Пример** |  |

* 1. Изменение специфики по ВИЧ

PUT api/MorbusHIV

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusHIV\_id (N, O) – Идентификатор * HIVContingentTypeP\_id (N, H) – Гражданство (значения 100 или 200 из справочника dbo.HIVContingentType) * Контингент: массив HIVContingentType\_Name\_list{ * HIVContingentType\_id (N,МH) – идентификатор контингента (справочник dbo.HIVContingentType)} * DiagD\_id (N,H) – Причина смерти * HIVDispOutCauseType\_id (N,H) – Причина снятия с диспансерного наблюдения (справочник dbo.HIVDispOutCauseType) * MorbusHIVOut\_endDT (D, H) – Дата снятия с диспансерного наблюдения * HIVInfectType\_id (N,H) – Тип вируса (справочник dbo.HIVInfectType) * HIVPregInfectStudyType\_id (N,H) – Стадия ВИЧ–инфекции (справочник dbo.HIVPregInfectStudyType) * HIVPregPathTransType\_id (N,H) – Предполагаемый путь инфицирования (спрвочник dbo.HIVPregPathTransType) * MorbusHIV\_confirmDate (D, Н) – Дата подтверждения диагноза (параметр обязательный, если диагноз из диапазона B20-B24 и HIVContingentTypeP\_id=100) * MorbusHIV\_EpidemCode (T, У) – Эпидемиологический код (параметр обязательный, если передан диагноз из диапазона B20-B24 и HIVContingentTypeP\_id=100) * --Данные раздела «Иммунный статус: CD4 Т-лимфоциты» * MorbusHIV\_CountCD4 (N, Н) – Количество CD4 Т–лимфоцитов (мм) * MorbusHIV\_PartCD4 (N, Н) – Процент содержания CD4 Т–лимфоцитов MorbusHIV\_DiagDT (D, Н) – Дата установления диагноза   --Данные раздела «Лабораторная диагностика ВИЧ-инфекции»:  --«Результат реакции иммуноблота»   * MorbusHIVLab\_BlotDT (D, H) – Дата иммунного блоттинга * MorbusHIV\_NumImmun (N, H) – номер иммуноблота * MorbusHIVLab\_BlotNum (T, H) – Номер серии результата реакции * MorbusHIVLab\_BlotResult (T, H) – Выявленные белки и гликопротеиды * MorbusHIVLab\_TestSystem (T, H) – Тип тест–системы * MorbusHIVLab\_id (N, H) – Идентификатор лабораторной диагностики * LabAssessmentResult\_iid (N, У) – Результат реакции иммуноблота (справочник dbo.LabAssessmentResult) (параметр обязательный, если передан MorbusHIVLab\_BlotDT)   --ИФА   * Lpu\_id (N, H) – Учреждение, первично выявившее положительный результат в ИФА * MorbusHIVLab\_IFADT (D, H) – Дата иммуноферментного анализа * MorbusHIVLab\_IFAResult (T, H) – Результат иммуноферментного анализа   --Полимеразная цепная реакци   * MorbusHIVLab\_PCRDT (D, H) – Дата полимеразной цепной реакции * MorbusHIVLab\_PCRResult (T, H) Результат полимеразной цепной реакции * LabAssessmentResult\_cid (N, У) – Результат полимеразной цепной реакции (справочник dbo.LabAssessmentResult) (параметр обязательный, если впервые передан MorbusHIVLab\_PCRDT) |
| **Ответ** | Успешный ответ с ошибкой «0» |

* 1. Получение специфики по ВИЧ

GET api/MorbusHIV

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Входящие параметры** | |  |  | | --- | --- | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  |  * PersonRegister\_id (N, Н) – Идентификатор записи регистра * MorbusHIV\_id (N, Н) – Идентификатор специфики |
| **Ответ** | Если не передан ни один параметр, то ошибка (не передан ни один из параметров).  Если не найдено ни одной записи, то ответ «0».  Если запись найдена, то успешный ответ – список параметров:   * MorbusHIV\_id (N, O) – Идентификатор * Person\_id (N, O) – Идентификатор человека * Diag\_id (N, O) – Идентификатор диагноза * HIVContingentTypeP\_id (N,H) – Гражданство (значения 100 или 200 из справочника dbo.HIVContingentType) * Контингент: массив HIVContingentType\_Name\_list{ * HIVContingentType\_id (N,МH) – идентификатор контингента (справочник dbo.HIVContingentType)} * DiagD\_id (N,H) – Причина смерти * HIVDispOutCauseType\_id (N,H) – Причина снятия с диспансерного наблюдения (справочник dbo.HIVDispOutCauseType) * MorbusHIVOut\_endDT (D, H) – Дата снятия с диспансерного наблюдения * HIVInfectType\_id (N,H) – Тип вируса (справочник dbo.HIVInfectType) * HIVPregInfectStudyType\_id (N,H) – Стадия ВИЧ–инфекции (справочник dbo.HIVPregInfectStudyType) * HIVPregPathTransType\_id (N,H) – Предполагаемый путь инфицирования (спрвочник dbo.HIVPregPathTransType) * MorbusHIV\_confirmDate (D, У) – Дата подтверждения диагноза (параметр обязательный, если диагноз из диапазона B20-B24 и HIVContingentTypeP\_id = 100) * MorbusHIV\_EpidemCode (T, У) – Эпидемиологический код (параметр обязательный, если диагноз из диапазона B20-B24 и HIVContingentTypeP\_id=100)   --Данные раздела «Иммунный статус: CD4 Т-лимфоциты»   * MorbusHIV\_CountCD4 (N, O) – Количество CD4 Т–лимфоцитов (мм) * MorbusHIV\_PartCD4 (N, O) – Процент содержания CD4 Т–лимфоцитов MorbusHIV\_DiagDT (D, O) – Дата установления диагноза   --Данные раздела «Лабораторная диагностика ВИЧ-инфекции»:  --«Результат реакции иммуноблота»   * MorbusHIVLab\_BlotDT (D, H) – Дата иммунного блоттинга * MorbusHIV\_NumImmun (N, H) – номер иммуноблота * MorbusHIVLab\_BlotNum (T, H) – Номер серии результата реакции * MorbusHIVLab\_BlotResult (T, H) – Выявленные белки и гликопротеиды * MorbusHIVLab\_TestSystem (T, H) – Тип тест–системы * MorbusHIVLab\_id (N, H) – Идентификатор лабораторной диагностики * LabAssessmentResult\_iid (N, У) – Результат реакции иммуноблота (справочник dbo.LabAssessmentResult) (параметр обязательный, если указан MorbusHIVLab\_BlotDT)   --ИФА   * Lpu\_id (N,H) – Учреждение, первично выявившее положительный результат в ИФА * MorbusHIVLab\_IFADT (D, H) – Дата иммуноферментного анализа * MorbusHIVLab\_IFAResult (T, H) – Результат иммуноферментного анализа   --Полимеразная цепная реакци   * MorbusHIVLab\_PCRDT (D, H) – Дата полимеразной цепной реакции * MorbusHIVLab\_PCRResult (T, H) Результат полимеразной цепной реакции * LabAssessmentResult\_cid (N, У) – Результат полимеразной цепной реакции (справочник dbo.LabAssessmentResult) (параметр обязательный, если указан MorbusHIVLab\_PCRDT) |
| **Пример** |  |

* 1. Создание химиопрофилактики ВИЧ–инфекции

Post api/MorbusHIVChem

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusHIV\_id (N, O) – Простое заболевание ВИЧ * Drug\_id (N, O) – Препарат в период проведения химиопрофилактики (справочник dbo.Drug) * MorbusHIVChem\_begDT (D, O) – Дата начала проведения * MorbusHIVChem\_Dose (T, O) – Доза препарата в период проведения химиопрофилактики * MorbusHIVChem\_endDT (D, H) – Дата окончания проведения |
| **Ответ** | **Успешный ответ идентификатор записи**   * MorbusHIVChem\_id (N, O) – идентификатор химиопрофилактики |
| **Пример** |  |

* 1. Получение химиопрофилактики ВИЧ–инфекции

GET api/MorbusHIVChem

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusHIVChem\_id (N, Н) – идентификатор химиопрофилактики * MorbusHIV (N, Н) – Заболевание ВИЧ |
| **Ответ** | Если не передан ни один параметр, то ошибка (не передан ни один из параметров).  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список с параметрами:   * MorbusHIVChem\_id (N, O) – идентификатор химиопрофилактики * MorbusHIV\_id (N, O) – Простое заболевание ВИЧ * Drug\_id (N, O) – Препарат в период проведения химиопрофилактики (справочник dbo.Drug) * MorbusHIVChem\_begDT (D, O) – Дата начала проведения * MorbusHIVChem\_Dose (T, O) – Доза препарата в период проведения химиопрофилактики * MorbusHIVChem\_endDT (D, H) – Дата окончания проведения |
| **Пример** |  |

* 1. Изменение химиопрофилактики ВИЧ–инфекции

PUT api/MorbusHIVChem

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusHIVChem\_id (N, O) – идентификатор химиопрофилактики * MorbusHIV\_id (N, Н) – Простое заболевание ВИЧ * Drug\_id (N, Н) – Препарат в период проведения химиопрофилактики (справочник dbo.Drug) * MorbusHIVChem\_begDT (D, Н) – Дата начала проведения * MorbusHIVChem\_Dose (T, Н) – Доза препарата в период проведения химиопрофилактики * MorbusHIVChem\_endDT (D, H) – Дата окончания проведения |
| **Ответ** | Успешный ответ с ошибкой «0» |
| **Пример** |  |

* 1. Создание вакцинации в рамках специфики ВИЧ

Post api/MorbusHIVVac

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusHIV\_id (N, O) – Простое заболевание ВИЧ * Drug\_id (N, O) – Препарат вакцинации (справочник dbo.Drug) * MorbusHIVVac\_setDT (D, О) – Дата вакцинации |
| **Ответ** | **Успешный ответ идентификатор записи**   * MorbusHIVVac\_id (N, O) – идентификатор вакцинации |
| **Пример** |  |

* 1. Получение вакцинации в рамках специфики ВИЧ

GET api/MorbusHIVVac

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusHIVVac\_id (N, Н) – идентификатор вакцинации * MorbusHIV (N, Н) – Заболевание ВИЧ |
| **Ответ** | Если не передан ни один параметр, то ошибка (не передан ни один из параметров).  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список с параметрами:   * MorbusHIVVac\_id (N, O) – идентификатор вакцинации * MorbusHIV\_id (N, O) – Простое заболевание ВИЧ * Drug\_id (N, O) – Препарат вакцинации (справочник dbo.Drug) * MorbusHIVVac\_setDT (D, О) – Дата вакцинации |
| **Пример** |  |

* 1. Изменение вакцинации в рамках специфики ВИЧ

PUT api/MorbusHIVVac

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusHIVVac\_id (N, O) – идентификатор вакцинации * MorbusHIV\_id (N, Н) – Простое заболевание ВИЧ * Drug\_id (N, Н) – Препарат вакцинации (справочник dbo.Drug) * MorbusHIVVac\_setDT (D, Н) – Дата вакцинации |
| **Ответ** | Успешный ответ с ошибкой «0» |
| **Пример** |  |

* 1. Создание вторичных заболеваний и оппортунистических инфекций в рамках специфики ВИЧ

Post api/MorbusHIVSecDiag

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusHIV\_id (N, O) – Простое заболевание ВИЧ * Diag\_id (N, O) – диагноз заболевания (справочник dbo.Drug) * MorbusHIVSecDiag\_setDT (D, О) – Дата установление |
| **Ответ** | **Успешный ответ идентификатор записи**   * MorbusHIVSecDiag\_id (N, O) – идентификатор |
| **Пример** |  |

* 1. Получение вторичных заболеваний и оппортунистических инфекций в рамках специфики ВИЧ

GET api/MorbusHIVSecDiag

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusHIVSecDiag\_id (N, Н) – идентификатор * MorbusHIV (N, Н) – Заболевание ВИЧ |
| **Ответ** | Если не передан ни один параметр, то ошибка (не передан ни один из параметров).  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список с параметрами:   * MorbusHIVSecDiag\_id (N, O) – идентификатор * MorbusHIV\_id (N, O) – Простое заболевание ВИЧ * Diag\_id (N, O) – диагноз заболевания (справочник dbo.Drug) * MorbusHIVSecDiag\_setDT (D, О) – Дата установление |
| **Пример** |  |

* 1. Изменение вторичных заболеваний и оппортунистических инфекций в рамках специфики ВИЧ

PUT api/MorbusHIVSecDiag

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusHIVSecDiag\_id (N, O) – идентификатор * MorbusHIV\_id (N, Н) – Простое заболевание ВИЧ * Diag\_id (N, Н) – диагноз заболевания (справочник dbo.Drug) * MorbusHIVSecDiag\_setDT (D, Н) – Дата установление |
| **Ответ** | Успешный ответ с ошибкой «0» |
| **Пример** |  |

* 1. Создание извещения о больном туберкулезом

POST api/EvnNotifyTub

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, O) – Идентификатор человека * PersonCategoryType\_id (N, O) – Категория населения справочник dbo.PersonCategoryType) * PersonLivingFacilies\_id (N, O) – Жилищные условия (справочник dbo.PersonLivingFacilies) * TubDetectionFactType\_id (N, O) – Обстоятельства, при которых выявлено заболевание (пути выявления) (справочник dbo.TubDetectionFactType) * EvnNotifyTub\_IsAutopsied (N, У) – Проводилась аутопсия, 2 – Да, 1 – Нет. (обязательно для заполнения, если TubDetectionFactType\_id=3) * EvnNotifyTub\_deathDT (Д, У) – Дата смерти (обязательно для заполнения, если TubDetectionFactType\_id=3) * TubResultDeathType\_id (N, У) – Причина смерти (справочник dbo.TubResultDeathType, обязательно для заполнения, если TubDetectionFactType\_id=3). * TubDetectionMethodType\_id (N, Н) – Выявлен из наблюдаемых в тубучреждениях групп (справочник dbo.TubDetectionMethodType) * TubDetectionMethodType\_id (N, O) – Метод выявления (справочник dbo.TubDetectionMethodType) * EvnNotifyTub\_OtherDetectionMethod (T, H) – Другой метод * DrugResistenceTest\_id (N, H) – Тестирование на лекарственную устойчивость (справочник dbo.DrugResistenceTest) * Diag\_id (N, O) – Код по МКБ–10 * TubDiagNotify\_id (N, O) –Диагноз (справочник dbo.TubDiagNotify) * TubDiagForm8\_id (N, O) – Заболевание по форме №8 (справочник dbo.TubDiagForm8) * EvnNotifyTub\_IsFirstDiag (N, H) – Установлен впервые в жизни (=2, если «да», =1, если «Нет») * TubClinicStateClass\_id (N, У) – Клиническая форм, обязательное для заполнения если в поле Diag\_id установлено одно из значений А15 (А15.0-А15.9), А16 (А16.0-А16.9). (справочник dbo.TubClinicStateClass) * TubLocalizationClass\_id (N, У) - Локализация туберкулеза, обязательное для заполнения если в поле Diag\_id установлено одно из значений А17 (А17.0, А17.1, А17.8, А17.9), А18 (А18.0-А18.9), А19 (А19.0-А19.2, А19.8, А19.9). (справочник dbo.TubLocalizationClass) * TubBacterialExcretion\_id (N, H) – Бактериовыделение (справочник dbo.TubBacterialExcretion) * TubMethodConfirmBactType\_id (N, H) – Метод подтверждения бактериовыделения (справочник dbo.TubMethodConfirmBactType) * TubRegCrazyType\_id (N, H) – Тип учета в наркологическом диспансере (справочник dbo.TubRegCrazyType) * TubDetectionPlaceType\_id (N, H) – Место выявления (справочник dbo.TubDetectionPlaceType) * EvnNotifyTub\_OtherDetectionPlace (T, H) – Учреждение EvnNotifyTub\_OtherPersonCategory (T, H) – Ведомство * PersonDecreedGroup\_id (N, O) – Декретированная группа (справочник dbo.PersonDecreedGroup) * EvnNotifyTub\_IsDecreeGroup (N, H) – Принадлежность к декретированным группам (=2, если «да», =1, если «Нет») * EvnNotifyTub\_IsDestruction (N, H) – Наличие распада (=2, если «да», =1, если «Нет») * EvnNotifyTub\_IsConfirmBact (N, H) – Подтверждение бактериовыделения (=2, если «да», =1, если «Нет») * EvnNotifyTub\_IsRegCrazy (N, H) – Состоит на учете в наркологическом диспансере (=2, если «да», =1, если «Нет») * EvnNotifyTub\_IsConfirmedDiag (N, O) – Диагноз подтвержден (=2, если «да», =1, если «Нет») * EvnNotifyTub\_DiagConfirmDT (D, H) – Дата подтверждения диагноза туберкулеза ЦВК * PersonDispGroup\_id (N, H) – Группа диспансерного наблюдения: (справочник dbo.PersonDispGroup) * EvnNotifyTub\_Comment (T, H) – Примечание * EvnNotifyTub\_FirstDT (D, H) – Дата первого обращения за мед/помощью * EvnNotifyTub\_RegDT (D, H) – Дата взятия на учет * EvnNotifyTub\_setDT (D, O) – Дата заполнения извещения * Lpu\_id (N, O) – МО заполнения извещения * MedPersonal\_id (N, O) – Врач, заполнивший извещение * TubFluorSurveyPeriodType\_id (N, H) – Сроки предыдущего ФГ обследования (справочник dbo.TubFluorSurveyPeriodType) * TubDiagSop\_id\_List (Т, H) – Сопутствующие заболевания (строка из отмеченных чекбоксами на форме значений TubDiagSop\_id через запятую, каждое значение записывается в dbo.TubDiagSopLink для данного извещения) * TubDiagSopLink\_Descr (T, H) – Прочие сопутствующие заболевания (записывается в dbo.TubDiagSopLink для данного извещения) * TubRiskFactorType\_id\_List (Т, H) – Фактор риска (строка из отмеченных чекбоксами на форме значений TubDiagFactorType\_id через запятую, каждое значение записывается в dbo.TubRiskFactorTypeLink для данного извещения) * Evn\_pid (N, О) – Идентификатор посещения/движения, в котором создается извещение * (D, У) – Дата разбора случая (обязательно для заполнения, если TubDetectionFactType\_id=3) * TubPostmortalDecision\_id (N, У) – Результат разбора случая (обязательно для заполнения, если TubDetectionFactType\_id=3) * TubSurveyGroupType\_id (N, H) – Выявлен из наблюдаемых в тубучреждениях групп (справочник dbo. TubSurveyGroupType)   --При автоматическом добавлении в регистр обязательны к заполнию поля:   * Lpu\_iid (N, У) – ЛПУ, добавившее человека в регистр * PersonRegister\_setDate (D, У) – Дата включения в регистр * MedPersonal\_iid (N, У) – врач, добавивший человека в регистр * PersonRegisterType\_id (N, У) – тип регистра (справочник dbo.PersonRegisterType) |
| **Ответ** | **Успешный ответ идентификатор записи**   * EvnNotifyTub\_id (N, O) – идентификатор извещения * PersonRegister\_id (N, У) – идентификатор записи регистра (при автоматическом добавлении в регистр).   Контроль:  Если у пациента уже существует извещение данного типа, то ошибка «Извещение по данному заболеванию существует». |
| **Пример** |  |

* 1. Получение извещения о больном туберкулезом

GET api/EvnNotifyTub

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnNotifyTub\_id (N, О) – идентификатор извещения * Person\_id (N, H) – Идентификатор человека |
| **Ответ** | Если не передан ни один параметр, то ошибка (не передан ни один из параметров).  Если не найдено ни одной записи, то ответ «0».  Если запись найдена, то передаются параметры:   * EvnNotifyTub\_id (N, О) – идентификатор извещения * Person\_id (N, O) – Идентификатор человека * PersonCategoryType\_id (N, O) – Категория населения справочник dbo.PersonCategoryType) * PersonLivingFacilies\_id (N, O) – Жилищные условия (справочник dbo.PersonLivingFacilies) * TubDetectionFactType\_id (N, O) – Обстоятельства, при которых выявлено заболевание (пути выявления) (справочник dbo.TubDetectionFactType) * EvnNotifyTub\_IsAutopsied (N, У) – Проводилась аутопсия, 2 – Да, 1 – Нет. (обязательно для заполнения, если TubDetectionFactType\_id=3) * EvnNotifyTub\_deathDT (Д, У) – Дата смерти (обязательно для заполнения, если TubDetectionFactType\_id=3) * TubResultDeathType\_id (N, У) – Причина смерти (справочник dbo.TubResultDeathType, обязательно для заполнения, если TubDetectionFactType\_id=3). * TubDetectionMethodType\_id (N, Н) – Выявлен из наблюдаемых в тубучреждениях групп (справочник dbo.TubDetectionMethodType) * TubDetectionMethodType\_id (N, O) – Метод выявления (справочник dbo.TubDetectionMethodType) * EvnNotifyTub\_OtherDetectionMethod (T, H) – Другой метод * DrugResistenceTest\_id (N, H) – Тестирование на лекарственную устойчивость (справочник dbo.DrugResistenceTest) * Diag\_id (N, O) – Код по МКБ–10 * TubDiagNotify\_id (N, O) –Диагноз (справочник dbo.TubDiagNotify) * TubDiagForm8\_id (N, O) – Заболевание по форме №8 (справочник dbo.TubDiagForm8) * EvnNotifyTub\_IsFirstDiag (N, H) – Установлен впервые в жизни (=2, если «да», =1, если «Нет») * TubClinicStateClass\_id (N, У) – Клиническая форм, обязательное для заполнения если в поле Diag\_id установлено одно из значений А15 (А15.0-А15.9), А16 (А16.0-А16.9). (справочник dbo.TubClinicStateClass) * TubLocalizationClass\_id (N, У) - Локализация туберкулеза, обязательное для заполнения если в поле Diag\_id установлено одно из значений А17 (А17.0, А17.1, А17.8, А17.9), А18 (А18.0-А18.9), А19 (А19.0-А19.2, А19.8, А19.9). (справочник dbo.TubLocalizationClass) * TubBacterialExcretion\_id (N, H) – Бактериовыделение (справочник dbo.TubBacterialExcretion) * TubMethodConfirmBactType\_id (N, H) – Метод подтверждения бактериовыделения (справочник dbo.TubMethodConfirmBactType) * TubRegCrazyType\_id (N, H) – Тип учета в наркологическом диспансере (справочник dbo.TubRegCrazyType) * TubDetectionPlaceType\_id (N, H) – Место выявления (справочник dbo.TubDetectionPlaceType) * EvnNotifyTub\_OtherDetectionPlace (T, H) – Учреждение EvnNotifyTub\_OtherPersonCategory (T, H) – Ведомство * PersonDecreedGroup\_id (N, O) – Декретированная группа (справочник dbo.PersonDecreedGroup) * EvnNotifyTub\_IsDecreeGroup (N, H) – Принадлежность к декретированным группам (=2, если «да», =1, если «Нет») * EvnNotifyTub\_IsDestruction (N, H) – Наличие распада (=2, если «да», =1, если «Нет») * EvnNotifyTub\_IsConfirmBact (N, H) – Подтверждение бактериовыделения (=2, если «да», =1, если «Нет») * EvnNotifyTub\_IsRegCrazy (N, H) – Состоит на учете в наркологическом диспансере (=2, если «да», =1, если «Нет») * EvnNotifyTub\_IsConfirmedDiag (N, O) – Диагноз подтвержден (=2, если «да», =1, если «Нет») * EvnNotifyTub\_DiagConfirmDT (D, H) – Дата подтверждения диагноза туберкулеза ЦВК * PersonDispGroup\_id (N, H) – Группа диспансерного наблюдения: (справочник dbo.PersonDispGroup) * EvnNotifyTub\_Comment (T, H) – Примечание * EvnNotifyTub\_FirstDT (D, H) – Дата первого обращения за мед/помощью * EvnNotifyTub\_RegDT (D, H) – Дата взятия на учет * EvnNotifyTub\_setDT (D, O) – Дата заполнения извещения * Lpu\_id (N, O) – МО заполнения извещения * MedPersonal\_id (N, O) – Врач, заполнивший извещение * TubFluorSurveyPeriodType\_id (N, H) – Сроки предыдущего ФГ обследования (справочник dbo.TubFluorSurveyPeriodType) * TubDiagSop\_id\_List (Т, H) – Сопутствующие заболевания (строка из значений TubDiagSop\_id через запятую из dbo.TubDiagSopLink для данного извещения) * TubDiagSopLink\_Descr (T, H) – Прочие сопутствующие заболевания (значение из в dbo.TubDiagSopLink для данного извещения) * TubRiskFactorType\_id\_List (Т, H) – Фактор риска (строка из значений TubDiagFactorType\_id через запятую из dbo.TubRiskFactorTypeLink для данного извещения) * Evn\_pid (N, У) – Идентификатор посещения/движения, в котором создано извещение * (D, У) – Дата разбора случая (обязательно для заполнения, если TubDetectionFactType\_id=3) * TubPostmortalDecision\_id (N, У) – Результат разбора случая (обязательно для заполнения, если TubDetectionFactType\_id=3) * TubSurveyGroupType\_id (N, H) – Выявлен из наблюдаемых в тубучреждениях групп (справочник dbo. TubSurveyGroupType)   --При автоматическом добавлении в регистр заполняются поля:   * Lpu\_iid (N, Н) – ЛПУ, добавившее человека в регистр * PersonRegister\_setDate (D, Н) – Дата включения в регистр * MedPersonal\_iid (N, Н) – врач, добавивший человека в регистр * PersonRegisterType\_id (N, Н) – тип регистра (справочник dbo.PersonRegisterType)   --Обязательные параметры при отказе включения в регистр:   * Lpu\_niid (N, У) – Лпу, не включившее в регистр * MedPersonal\_niid (N, У) – Врач, не включивший в регистр * PersonRegisterFailIncludeCause\_id (N, У) – Причина невключения в регистр (справочник dbo.PersonRegisterFailIncludeCause) |
| **Пример** |  |

* 1. Изменение извещения о больном туберкулезом

PUT api/EvnNotifyTub

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnNotifyTub\_id (N, О) – идентификатор извещения * Person\_id (N, Н) – Идентификатор человека * PersonCategoryType\_id (N, Н) – Категория населения справочник dbo.PersonCategoryType) * PersonLivingFacilies\_id (N, Н) – Жилищные условия (справочник dbo.PersonLivingFacilies) * TubDetectionFactType\_id (N, Н) – Обстоятельства, при которых выявлено заболевание (пути выявления) (справочник dbo.TubDetectionFactType) * EvnNotifyTub\_IsAutopsied (N, У) – Проводилась аутопсия, 2 – Да, 1 – Нет. (обязательно для заполнения, если изменился параметр TubDetectionFactType\_id=3) * EvnNotifyTub\_deathDT (Д, У) – Дата смерти (обязательно для заполнения, если изменился параметр TubDetectionFactType\_id=3) * TubResultDeathType\_id (N, У) – Причина смерти (справочник dbo.TubResultDeathType, обязательно для заполнения, если изменился параметр TubDetectionFactType\_id=3). * TubDetectionMethodType\_id (N, Н) – Выявлен из наблюдаемых в тубучреждениях групп (справочник dbo.TubDetectionMethodType) * TubDetectionMethodType\_id (N, Н) – Метод выявления (справочник dbo.TubDetectionMethodType) * EvnNotifyTub\_OtherDetectionMethod (T, H) – Другой метод * DrugResistenceTest\_id (N, H) – Тестирование на лекарственную устойчивость (справочник dbo.DrugResistenceTest) * Diag\_id (N, Н) – Код по МКБ–10 * TubDiagNotify\_id (N, Н) – Диагноз (справочник dbo.TubDiagNotify) * TubDiagForm8\_id (N, Н) – Заболевание по форме №8 (справочник dbo.TubDiagForm8) * EvnNotifyTub\_IsFirstDiag (N, H) – Установлен впервые в жизни (=2, если «да», =1, если «Нет») * TubClinicStateClass\_id (N, У) – Клиническая форма, обязательное для заполнения, если изменено значение в поле Diag\_id на: А15 (А15.0-А15.9), А16 (А16.0-А16.9). (справочник dbo.TubClinicStateClass) * TubLocalizationClass\_id (N, У) - Локализация туберкулеза, обязательное для заполнения, если изменено значение в поле Diag\_id на: А17 (А17.0, А17.1, А17.8, А17.9), А18 (А18.0-А18.9), А19 (А19.0-А19.2, А19.8, А19.9). (справочник dbo.TubLocalizationClass) * TubBacterialExcretion\_id (N, H) – Бактериовыделение (справочник dbo.TubBacterialExcretion) * TubMethodConfirmBactType\_id (N, H) – Метод подтверждения бактериовыделения (справочник dbo.TubMethodConfirmBactType) * TubRegCrazyType\_id (N, H) – Тип учета в наркологическом диспансере (справочник dbo.TubRegCrazyType) * TubDetectionPlaceType\_id (N, H) – Место выявления (справочник dbo.TubDetectionPlaceType) * EvnNotifyTub\_OtherDetectionPlace (T, H) – Учреждение EvnNotifyTub\_OtherPersonCategory (T, H) – Ведомство * PersonDecreedGroup\_id (N, Н) – Декретированная группа (справочник dbo.PersonDecreedGroup) * EvnNotifyTub\_IsDecreeGroup (N, H) – Принадлежность к декретированным группам (=2, если «да», =1, если «Нет») * EvnNotifyTub\_IsDestruction (N, H) – Наличие распада (=2, если «да», =1, если «Нет») * EvnNotifyTub\_IsConfirmBact (N, H) – Подтверждение бактериовыделения (=2, если «да», =1, если «Нет») * EvnNotifyTub\_IsRegCrazy (N, H) – Состоит на учете в наркологическом диспансере (=2, если «да», =1, если «Нет») * EvnNotifyTub\_IsConfirmedDiag (N, Н) – Диагноз подтвержден (=2, если «да», =1, если «Нет») * EvnNotifyTub\_DiagConfirmDT (D, H) – Дата подтверждения диагноза туберкулеза ЦВК * PersonDispGroup\_id (N, H) – Группа диспансерного наблюдения: (справочник dbo.PersonDispGroup) * EvnNotifyTub\_Comment (T, H) – Примечание * EvnNotifyTub\_FirstDT (D, H) – Дата первого обращения за мед/помощью * EvnNotifyTub\_RegDT (D, H) – Дата взятия на учет * EvnNotifyTub\_setDT (D, Н) – Дата заполнения извещения * Lpu\_id (N, Н) – МО заполнения извещения * MedPersonal\_id (N, Н) – Врач, заполнивший извещение * TubFluorSurveyPeriodType\_id (N, H) – Сроки предыдущего ФГ обследования (справочник dbo.TubFluorSurveyPeriodType) * TubDiagSop\_id\_List (Т, H) – Сопутствующие заболевания (строка из отмеченных чекбоксами на форме значений TubDiagSop\_id через запятую, каждое значение записывается в dbo.TubDiagSopLink для данного извещения, если для данного извещения были значения, не возвращенные в строке, то они удаляются) * TubDiagSopLink\_Descr (T, H) – Прочие сопутствующие заболевания (записывается в dbo.TubDiagSopLink для данного извещения) * TubRiskFactorType\_id\_List (Т, H) – Фактор риска (строка из отмеченных чекбоксами на форме значений TubDiagFactorType\_id через запятую, каждое значение записывается в dbo.TubRiskFactorTypeLink для данного извещения, если для данного извещения были значения, не возвращенные в строке, то они удаляются) * (D, У) – Дата разбора случая (обязательно для заполнения, если TubDetectionFactType\_id=3). * TubPostmortalDecision\_id (N, У) – Результат разбора случая (обязательно для заполнения, если TubDetectionFactType\_id=3) * TubSurveyGroupType\_id (N, H) – Выявлен из наблюдаемых в тубучреждениях групп (справочник dbo. TubSurveyGroupType)   --Обязательные параметры при отказе включения в регистр:   * Lpu\_niid (N, У) – Лпу, не включившее в регистр * MedPersonal\_niid (N, У) – Врач, не включивший в регистр * PersonRegisterFailIncludeCause\_id (N, У) – Причина невключения в регистр (справочник dbo.PersonRegisterFailIncludeCause) |
| **Ответ** | Успешный ответ с ошибкой «0» |
| **Пример** |  |

* 1. Получение специфики по туберкулезу

GET api/MorbusTub

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTub\_id (N, O) – идентификатор специфики |
| **Ответ** | * PersonRegister\_id (N, Н) – идентификатор записи регистра * MorbusTub\_id (N, O) – идентификатор специфики * PersonRegister\_id (N, H) – Запись регистра * Person\_id (N, H) – Идентификатор человека * PersonResidenceType\_id (N, H) – Статус пациента (справочник dbo.PersonResidenceType) * PersonDecreedGroup\_id (N, H) – Декретированная группа (справочник dbo.PersonDecreedGroup) * PersonLivingFacilies\_id (N, H) – Жилищные условия (справочник dbo.PersonLivingFacilies) * Diag\_id (N, О) – Диагноз МКБ–10 * TubDiag\_id (N, H) – Диагноз * TubDiagSop\_id\_List (T, Н) – Сопутствующие заболевания (строка из значений TubDiagSop\_id через запятую из dbo.TubDiagSopLink для данной специфики) * TubDiagSopLink\_Descr (T, H) – Прочие сопутствующие заболевания (значение из в dbo.TubDiagSopLink для данной специфики) * TubRiskFactorType\_id\_List (T, H) – Фактор риска (строка из значений TubDiagFactorType\_id через запятую из dbo.TubRiskFactorTypeLink для данной специфики) * Evn\_pid (N, Н) – Идентификатор посещения/движения, в котором заведена специфика * TubSickGroupType\_id (N, H) – Группа больных (справочник dbo.TubSickGroupType) * MorbusTub\_begDT (D, H) – Дата возникновения симптомов * MorbusTub\_FirstDT (D, H) – Дата первого обращения к любому врачу по поводу этих симптомов * MorbusTub\_DiagDT (D, H) – Дата установления диагноза * TubResultChemType\_id (N, H) – Тип исхода курса химиотерапии (справочник dbo.TubResultChemType) * TubResultChemType\_id (N, H) – Тип подтверждения (справочник dbo. TubResultChemType) * MorbusTub\_ResultDT (D, H) – Дата исхода курса химиотерапии * TubResultDeathType\_id (N, H) – Причина смерти (справочник dbo. TubResultDeathType) * MorbusTub\_deadDT (D, H) – Дата смерти * MorbusTub\_breakDT (D, H) – Дата прерывания курса химиотерапии * TubBreakChemType\_id (N, H) – Причина прерывания химиотерапии (справочник dbo. TubBreakChemType) * MorbusTub\_disDT (D, H) – Выбыл * PersonDispGroup\_id (N, H) – Группа диспансерного наблюдения (справочник dbo. PersonDispGroup) * MorbusTub\_ConvDT (D, H) – Дата перевода в III группу ДУ * MorbusTub\_unsetDT (D, H) – Дата снятия диагноза туберкулеза * MorbusTub\_CountDay (N, H) – Общее кол–во дней нетрудоспособности |
| **Пример** |  |

* 1. Изменение специфики по туберкулезу

PUT api/MorbusTub

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTub\_id (N, O) – идентификатор специфики * PersonRegister\_id (N, H) – Запись регистра * Person\_id (N, H) – Идентификатор человека * PersonResidenceType\_id (N, H) – Статус пациента (справочник dbo.PersonResidenceType) * PersonDecreedGroup\_id (N, H) – Декретированная группа (справочник dbo.PersonDecreedGroup) * PersonLivingFacilies\_id (N, H) – Жилищные условия (справочник dbo.PersonLivingFacilies) * Diag\_id (N, Н) – Диагноз МКБ–10 * TubDiag\_id (N, H) – Диагноз * TubDiagSop\_id\_Link (Т, H) – Сопутствующие заболевания (строка из отмеченных чекбоксами на форме значений TubDiagSop\_id через запятую, каждое значение записывается в dbo.TubDiagSopLink для данной специфики, если для данной специфики были значения, не возвращенные в строке, то они удаляются) * TubDiagSopLink\_Descr (T, H) – Прочие сопутствующие заболевания (записывается в dbo.TubDiagSopLink для данной специфики) * TubRiskFactorType\_id\_Link (T, H) – Фактор риска (строка из отмеченных чекбоксами на форме значений TubDiagFactorType\_id через запятую, каждое значение записывается в dbo.TubRiskFactorTypeLink для данной специфики, если для данной специфики были значения, не возвращенные в строке, то они удаляются) * TubSickGroupType\_id (N, H) – Группа больных (справочник dbo.TubSickGroupType) * MorbusTub\_begDT (D, H) – Дата возникновения симптомов * MorbusTub\_FirstDT (D, H) – Дата первого обращения к любому врачу по поводу этих симптомов * MorbusTub\_DiagDT (D, H) – Дата установления диагноза * TubResultChemType\_id (N, H) – Тип исхода курса химиотерапии (справочник dbo.TubResultChemType) * TubResultChemType\_id (N, H) – Тип подтверждения (справочник dbo. TubResultChemType) * MorbusTub\_ResultDT (D, H) – Дата исхода курса химиотерапии * TubResultDeathType\_id (N, H) – Причина смерти (справочник dbo. TubResultDeathType) * MorbusTub\_deadDT (D, H) – Дата смерти * MorbusTub\_breakDT (D, H) – Дата прерывания курса химиотерапии * TubBreakChemType\_id (N, H) – Причина прерывания химиотерапии (справочник dbo. TubBreakChemType) * MorbusTub\_disDT (D, H) – Выбыл * PersonDispGroup\_id (N, H) – Группа диспансерного наблюдения (справочник dbo. PersonDispGroup) * MorbusTub\_ConvDT (D, H) – Дата перевода в III группу ДУ * MorbusTub\_unsetDT (D, H) – Дата снятия диагноза туберкулеза * MorbusTub\_CountDay (N, H) – Общее кол–во дней нетрудоспособности |
| **Ответ** | Успешный ответ с ошибкой «0» |
| **Пример** |  |

* 1. Создание записи Генерализированные формы в рамках специфики по туберкулезу

POST api/TubDiagGeneralForm

|  |  |
| --- | --- |
| **Входящие параметры** | * Diag\_id (N, O) – Диагноз * MorbusTub\_id (N, O) – Специфика по туберкулезу * TubDiagGeneralForm\_setDT (N, Н) – Дата выявления |
| **Ответ** | **Успешный ответ идентификатор записи**   * TubDiagGeneralForm\_id (N, O) – идентификатор записи |
| **Пример** |  |

* 1. Получение записи Генерализированные формы в рамках специфики по туберкулезу

GET api/TubDiagGeneralForm

|  |  |
| --- | --- |
| **Входящие параметры** | * TubDiagGeneralForm\_id (N, Н) – идентификатор записи * MorbusTub\_id (N, Н) – Специфика по туберкулезу |
| **Ответ** | Если не передан ни один параметр, то ошибка (не передан ни один из параметров).  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список с параметрами:   * TubDiagGeneralForm\_id (N, Н) – идентификатор записи * Diag\_id (N, O) – Диагноз * TubDiagGeneralForm\_setDT (N, Н) – Дата выявления * MorbusTub\_id (N, Н) – Специфика по туберкулезу |
| **Пример** |  |

* 1. Изменение записи Генерализированные формы в рамках специфики по туберкулезу

PUT api/TubDiagGeneralForm

|  |  |
| --- | --- |
| **Входящие параметры** | * TubDiagGeneralForm\_id (N, O) – идентификатор записи * Diag\_id (N, Н) – Диагноз * MorbusTub\_id (N, Н) – Специфика по туберкулезу * TubDiagGeneralForm\_setDT (D, Н) – Дата выявления |
| **Ответ** | Успешный ответ с ошибкой «0» |

* 1. Создание режима химиотерапии в рамках специфики по туберкулезу

POST api/ MorbusTubConditChem

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTub\_id (N, O) – Простое заболевание (туберкулез) * MorbusTubConditChem\_BegDate (D, O) – Дата начала лечения * MorbusTubConditChem\_EndDate (D, Н) – Дата окончания лечения * TubStageChemType\_id (N, Н) – Фазы химиотерапии (справочник dbo. TubStageChemType) * TubStandartConditChemType\_id (N, O) – Стандартные режимы химиотерапии (справочник dbo. TubStandartConditChemType) * TubVenueType\_id (N, O) – Тип места проведения (справочник dbo. TubVenueType) |
| **Ответ** | **Успешный ответ идентификатор записи**   * TubDiagGeneralForm\_id (N, O) – идентификатор записи |

* 1. Получение режима химиотерапии в рамках специфики по туберкулезу

GET api/ MorbusTubConditChem

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubConditChem\_id (N, Н) – идентификатор записи * MorbusTub\_id (N, Н) – Простое заболевание (туберкулез) |
| **Ответ** | Если не передан ни один параметр, то ошибка (не передан ни один из параметров).  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список с параметрами:   * MorbusTub\_id (N, O) – Простое заболевание (туберкулез) * MorbusTubConditChem\_BegDate (D, O) – Дата начала лечения * MorbusTubConditChem\_EndDate (D, Н) – Дата окончания лечения * TubStageChemType\_id (N, O) – Фазы химиотерапии (справочник dbo. TubStageChemType) * TubStandartConditChemType\_id (N, O) – Стандартные режимы химиотерапии (справочник dbo. TubStandartConditChemType) * TubVenueType\_id (N, O) – Тип места проведения (справочник dbo. TubVenueType) |

* 1. Изменение режима химиотерапии в рамках специфики по туберкулезу

PUT api/MorbusTubConditChem

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubConditChem\_id (N, O) – идентификатор записи * MorbusTub\_id (N, Н) – Простое заболевание (туберкулез) * MorbusTubConditChem\_BegDate (D, Н) – Дата начала лечения * MorbusTubConditChem\_EndDate (D, Н) – Дата окончания лечения * TubStageChemType\_id (N, Н) – Фазы химиотерапии (справочник dbo. TubStageChemType) * TubStandartConditChemType\_id (N, Н) – Стандартные режимы химиотерапии (справочник dbo. TubStandartConditChemType) * TubVenueType\_id (N, Н) – Тип места проведения (справочник dbo. TubVenueType) |
| **Ответ** | Успешный ответ с ошибкой «0» |

* 1. Создание консультации фтизиохирурга в рамках специфики по туберкулезу

POST api/MorbusTubAdvice

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTub\_id (N, O) – Простое заболевание: Туберкулез * MorbusTubAdvice\_setDT (D, O) –Дата проведения * TubAdviceResultType\_id (N, O) – Результат консультации (справочник dbo. TubAdviceResultType) |
| **Ответ** | **Успешный ответ идентификатор записи**   * MorbusTubAdvice\_id (N, O) – идентификатор |

* 1. Получение консультации фтизиохирурга в рамках специфики по туберкулезу

GET api/MorbusTubAdvice

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubAdvice\_id (N, Н) – идентификатор * MorbusTub\_id (N, Н) – Простое заболевание (туберкулез) |
| **Ответ** | Если не передан ни один параметр, то ошибка (не передан ни один из параметров).  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список с параметрами:   * MorbusTubAdvice\_id (N, O) – идентификатор * MorbusTub\_id (N, O) – Простое заболевание: Туберкулез * MorbusTubAdvice\_setDT (D, O) –Дата проведения * TubAdviceResultType\_id (N, O) – Результат консультации (справочник dbo. TubAdviceResultType) |

* 1. Изменение консультации фтизиохирурга в рамках специфики по туберкулезу

PUT api/MorbusTubAdvice

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubAdvice\_id (N, O) – идентификатор * MorbusTub\_id (N, Н) – Простое заболевание: Туберкулез * MorbusTubAdvice\_setDT (D, Н) –Дата проведения * TubAdviceResultType\_id (N, Н) – Результат консультации (справочник dbo. TubAdviceResultType) |
| **Ответ** | Успешный ответ с ошибкой «0» |

* 1. Создание записи об оперативном лечении в рамках специфики по туберкулезу

POST api/MorbusTubAdviceOper

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTub\_id (N, O) – Простое заболевание: Туберкулез * MorbusTubAdvice\_id (N, O) – Консультация фтизиохирурга * MorbusTubAdviceOper\_setDT (D, O) –Дата операции * UslugaComplex\_id (N, O) – Тип операции (справочник dbo.UslugaComplex) |
| **Ответ** | **Успешный ответ идентификатор записи**   * MorbusTubAdviceOper\_id (N, O) – идентификатор |

* 1. Получение записи об оперативном лечении в рамках специфики по туберкулезу

GET api/MorbusTubAdviceOper

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubAdviceOper\_id (N, Н) – идентификатор * MorbusTub\_id (N, Н) – Простое заболевание: Туберкулез * MorbusTubAdvice\_id (N, Н) – Консультация фтизиохирурга |
| **Ответ** | Если не передан ни один параметр, то ошибка (не передан ни один из параметров).  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список с параметрами:   * MorbusTubAdviceOper\_id (N, O) – идентификатор * MorbusTub\_id (N, O) – Простое заболевание: Туберкулез * MorbusTubAdvice\_id (N, O) – Консультация фтизиохирурга * MorbusTubAdviceOper\_setDT (D, O) –Дата операции * UslugaComplex\_id (N, O) – Тип операции (справочник dbo.UslugaComplex) |

* 1. Изменение записи об оперативном лечении в рамках специфики по туберкулезу

PUT api/MorbusTubAdviceOper

|  |  |
| --- | --- |
| **Входящие параметры** | MorbusTubAdviceOper\_id (N, O) – идентификатор  MorbusTub\_id (N, Н) – Простое заболевание: Туберкулез  MorbusTubAdvice\_id (N, Н) – Консультация фтизиохирурга  MorbusTubAdviceOper\_setDT (D, Н) – Дата операции  UslugaComplex\_id (N, Н) – Тип операции (справочник dbo.UslugaComplex) |
| **Ответ** | Успешный ответ с ошибкой «0» |

* 1. Создание направления на проведение микроскопических исследований на туберкулез

POST api/EvnDirectionTub

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTub\_id (N, O) – Простое заболевание: Туберкулез * EvnDirectionTub\_setDT (D, O) – Дата направления * TubDiagnosticMaterialType\_id (N, O) – Диагностический материал (справочник dbo.TubDiagnosticMaterialType) * TubTargetStudyType\_id (N, O) – Цель исследования (справочник dbo.TubTargetStudyType) * EvnDirectionTub\_PersonRegNum (T, H) – Региональный регистрационный номер пациента * MedPersonal\_id (N, O) – Направивший медработник * MedPersonal\_lid (N, O) – Медицинский работник, собравший образцы диагностического материала * EvnDirectionTub\_NumLab (T, H) – Лабораторный номер исследования * EvnDirectionTub\_ResDT (D, Н) – Дата выдачи результата |
| **Ответ** | **Успешный ответ идентификатор записи**   * EvnDirectionTub\_id (N, O) – Идентификатор |

* 1. Получение направления на проведение микроскопических исследований на туберкулез

GET api/EvnDirectionTub

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnDirectionTub\_id (N, Н) – Идентификатор * MorbusTub\_id (N, Н) – Простое заболевание: Туберкулез |
| **Ответ** | Если не передан ни один параметр, то ошибка (не передан ни один из параметров).  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список с параметрами:   * EvnDirectionTub\_id (N, O) – Идентификатор * MorbusTub\_id (N, O) – Простое заболевание: Туберкулез * EvnDirectionTub\_setDT (D, O) – Дата направления * TubDiagnosticMaterialType\_id (N, O) – Диагностический материал (справочник dbo.TubDiagnosticMaterialType) * TubTargetStudyType\_id (N, O) – Цель исследования (справочник dbo.TubTargetStudyType) * EvnDirectionTub\_PersonRegNum (T, H) – Региональный регистрационный номер пациента * MedPersonal\_id (N, O) – Направивший медработник * MedPersonal\_lid (N, O) – Медицинский работник, собравший образцы диагностического материала * EvnDirectionTub\_NumLab (T, H) – Лабораторный номер исследования * EvnDirectionTub\_ResDT (D, Н) – Дата выдачи результата |

* 1. Изменение направления на проведение микроскопических исследований на туберкулез

PUT api/EvnDirectionTub

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnDirectionTub\_id (N, O) – Идентификатор * MorbusTub\_id (N, Н) – Простое заболевание: Туберкулез * EvnDirectionTub\_setDT (D, Н) – Дата направления * TubDiagnosticMaterialType\_id (N, Н) – Диагностический материал (справочник dbo.TubDiagnosticMaterialType) * TubTargetStudyType\_id (N, Н) – Цель исследования (справочник dbo.TubTargetStudyType) * EvnDirectionTub\_PersonRegNum (T, H) – Региональный регистрационный номер пациента * MedPersonal\_id (N, Н) – Направивший медработник * MedPersonal\_lid (N, Н) – Медицинский работник, собравший образцы диагностического материала * EvnDirectionTub\_NumLab (T, H) – Лабораторный номер исследования * EvnDirectionTub\_ResDT (D, Н) – Дата выдачи результата |
| **Ответ** | Успешный ответ с ошибкой «0» |

* 1. Создание результатов микроскопических исследований в рамках специфики по туберкулезу

POST api/TubMicrosResult

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnDirectionTub\_id (N, O) – Идентификатор направления на проведение микроскопических исследований на туберкулез * TubMicrosResult\_MicrosDT (D, O) – Дата исследования * TubMicrosResult\_Num (N, Н) – № образца мокроты * TubMicrosResult\_setDT (D, O) – Дата сбора образцов * TubMicrosResultType\_id (N, O) – Результат микроскопии (справочник dbo.TubMicrosResultType) * TubMicrosResult\_EdResult (N, Н) – Число микобактерий TubMicrosResult\_Comment (T, H) – Примечание |
| **Ответ** | **Успешный ответ идентификатор записи**   * TubMicrosResult\_id (N, O) – Идентификатор |

* 1. Получение результатов микроскопических исследований в рамках специфики по туберкулезу

GET api/TubMicrosResult

|  |  |
| --- | --- |
| **Входящие параметры** | * TubMicrosResult\_id (N, Н) – Идентификатор * EvnDirectionTub\_id (N, Н) – Идентификатор направления на проведение микроскопических исследований на туберкулез |
| **Ответ** | Если не передан ни один параметр, то ошибка (не передан ни один из параметров).  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список с параметрами:   * EvnDirectionTub\_id (N, O) – Идентификатор направления на проведение микроскопических исследований на туберкулез * TubMicrosResult\_id (N, Н) – Идентификатор * TubMicrosResult\_MicrosDT (D, O) – Дата исследования * TubMicrosResult\_Num (N, Н) – № образца мокроты * TubMicrosResult\_setDT (D, O) – Дата сбора образцов * TubMicrosResultType\_id (N, O) – Результат микроскопии (справочник dbo.TubMicrosResultType) * TubMicrosResult\_EdResult (N, Н) – Число микобактерий TubMicrosResult\_Comment (T, H) – Примечание |

* 1. Изменение результатов микроскопических исследований в рамках специфики по туберкулезу

PUT api/TubMicrosResult

|  |  |
| --- | --- |
| **Входящие параметры** | * TubMicrosResult\_id (N, О) – Идентификатор * EvnDirectionTub\_id (N, Н) – Идентификатор направления на проведение микроскопических исследований на туберкулез * TubMicrosResult\_MicrosDT (D, Н) – Дата исследования * TubMicrosResult\_Num (N, Н) – № образца мокроты * TubMicrosResult\_setDT (D, Н) – Дата сбора образцов * TubMicrosResultType\_id (N, Н) – Результат микроскопии (справочник dbo.TubMicrosResultType) * TubMicrosResult\_EdResult (N, Н) – Число микобактерий TubMicrosResult\_Comment (T, H) – Примечание |
| **Ответ** | Успешный ответ с ошибкой «0» |

* 1. Создание лекарственных назначений в рамках специфики по туберкулезу

POST api/MorbusTubPrescr

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTub\_id (N, O) – Простое заболевание: Туберкулез * TubStageChemType\_id (N, O) – Фаза химиотерапии (справочник dbo.TubStageChemType) * MorbusTubPrescr\_setDT (D, О) – Дата назначения * MorbusTubPrescr\_endDate (D, H) – Дата отмены * TubDrug\_id (N, У) – Препарат (справочник dbo.TubDrug), обязательный, если не указан Drug\_id * Drug\_id (N, У) – Препарат (РЛС) (справочник dbo.Drug), обязательный, если не указан TubDrug\_id * MorbusTubPrescr\_DoseDay (T, Н) – Суточная доза * MorbusTubPrescr\_Schema (T, Н) – Схема * MorbusTubPrescr\_DoseTotal (T, Н) – Общее кол. Доз |
| **Ответ** | **Успешный ответ идентификатор записи**  MorbusTubPrescr\_id – идентификатор записи |

* 1. Получение лекарственных назначений в рамках специфики по туберкулезу

GET api/MorbusTubPrescr

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubPrescr\_id (N, Н) – идентификатор записи * MorbusTub\_id (N, Н) – Простое заболевание: Туберкулез |
| **Ответ** | Если не передан ни один параметр, то ошибка (не передан ни один из параметров).  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список с параметрами:   * MorbusTub\_id (N, O) – Простое заболевание: Туберкулез * MorbusTubPrescr\_id (N, О) – идентификатор записи * TubStageChemType\_id (N, O) – Фаза химиотерапии (справочник dbo.TubStageChemType) * MorbusTubPrescr\_setDT (D, О) – Дата назначения * MorbusTubPrescr\_endDate (D, H) – Дата отмены * TubDrug\_id (N, У) – Препарат (справочник dbo.TubDrug), обязательный, если не указан Drug\_id * Drug\_id (N, У) – Препарат (РЛС) (справочник dbo.Drug), обязательный, если не указан TubDrug\_id * MorbusTubPrescr\_DoseDay (T, Н) – Суточная доза * MorbusTubPrescr\_Schema (T, Н) – Схема * MorbusTubPrescr\_DoseTotal (T, Н) – Общее кол. Доз |

* 1. Изменение лекарственных назначений в рамках специфики по туберкулезу

PUT api/MorbusTubPrescr

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubPrescr\_id (N, О) – идентификатор записи * MorbusTub\_id (N, Н) – Простое заболевание: Туберкулез * TubStageChemType\_id (N, Н) – Фаза химиотерапии (справочник dbo.TubStageChemType) * MorbusTubPrescr\_setDT (D, Н) – Дата назначения * MorbusTubPrescr\_endDate (D, H) – Дата отмены * TubDrug\_id (N, У) – Препарат (справочник dbo.TubDrug), обязательный, если не указан Drug\_id * Drug\_id (N, У) – Препарат (РЛС) (справочник dbo.Drug), обязательный, если не указан TubDrug\_id * MorbusTubPrescr\_DoseDay (T, Н) – Суточная доза * MorbusTubPrescr\_Schema (T, Н) – Схема * MorbusTubPrescr\_DoseTotal (T, Н) – Общее кол. Доз |
| **Ответ** | Успешный ответ с ошибкой «0» |

* 1. Создание графика исполнения назначения процедур в рамках специфики по туберкулезу

POST api/MorbusTubPrescrTimetable

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubPrescr\_id (N, O) – ссылка на лекарственное назначение MorbusTubPrescrTimetable\_setDT (D, О) – Дата * MorbusTubPrescrTimetable\_IsExec (N, Н) – отметка о выполнении (=2, если «Да», =1, если «Нет») * MedPersonal\_id (N, O) – врач–пользователь, выполнивший назначение |
| **Ответ** | **Успешный ответ идентификатор записи**   * MorbusTubPrescrTimetable\_id – идентификатор записи |

* 1. Получение графика исполнения назначения процедур в рамках специфики по туберкулезу

GET api/MorbusTubPrescrTimetable

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubPrescrTimetable\_id (N, Н) – идентификатор записи * MorbusTubPrescr\_id (N, Н) – ссылка на лекарственное назначение |
| **Ответ** | Если не передан ни один параметр, то ошибка (не передан ни один из параметров).  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список с параметрами:   * MorbusTubPrescrTimetable\_id (N, О) – идентификатор записи * MorbusTubPrescr\_id (N, O) – ссылка на лекарственное назначение MorbusTubPrescrTimetable\_setDT (D, О) – Дата * MorbusTubPrescrTimetable\_IsExec (N, Н) – отметка о выполнении (=2, если «Да», =1, если «Нет») * MedPersonal\_id (N, O) – врач–пользователь, выполнивший назначение |

* 1. Изменение графика исполнения назначения процедур в рамках специфики по туберкулезу

PUT api/MorbusTubPrescrTimetable

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubPrescrTimetable\_id (N, О) – идентификатор записи * MorbusTubPrescr\_id (N, Н) – ссылка на лекарственное назначение MorbusTubPrescrTimetable\_setDT (D, Н) – Дата * MorbusTubPrescrTimetable\_IsExec (N, Н) – отметка о выполнении (=2, если «Да», =1, если «Нет») * MedPersonal\_id (N, Н) – врач–пользователь, выполнивший назначение |
| **Ответ** | Успешный ответ с ошибкой «0» |

* 1. Создание результатов исследования в рамках специфики по туберкулезу

POST api/MorbusTubStudyResult

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTub\_id (N, O) – Простое заболевание: Туберкулез * PersonWeight\_id (N, H) – Идентификатор веса * PersonWeight\_Weight (N, H) – Вес * TubStageChemType\_id (N, О) – Фазы химиотерапии (справочник dbo.TubStageChemType) |
| **Ответ** | **Успешный ответ идентификатор записи**   * MorbusTubStudyResult\_id (N, O) – Результаты исследований |

* 1. Получение результатов исследования в рамках специфики по туберкулезу

GET api/MorbusTubStudyResult

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubStudyResult\_id (N, Н) – Результаты исследований * MorbusTub\_id (N, Н) – Простое заболевание: Туберкулез |
| **Ответ** | Если не передан ни один параметр, то ошибка (не передан ни один из параметров).  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список с параметрами:   * MorbusTubStudyResult\_id (N, O) – Результаты исследований * MorbusTub\_id (N, O) – Простое заболевание: Туберкулез * PersonWeight\_id (N, H) – Идентификатор веса * PersonWeight\_Weight (N, H) – Вес * TubStageChemType\_id (N, О) – Фазы химиотерапии (справочник dbo.TubStageChemType) |

* 1. Изменение результатов исследования в рамках специфики по туберкулезу

PUT api/MorbusTubStudyResult

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubStudyResult\_id (N, O) – Результаты исследований * MorbusTub\_id (N, Н) – Простое заболевание: Туберкулез * PersonWeight\_id (N, H) – Идентификатор веса * PersonWeight\_Weight (N, H) – Вес * TubStageChemType\_id (N, Н) – Фазы химиотерапии (справочник dbo.TubStageChemType) |
| **Ответ** | Успешный ответ с ошибкой «0» |

* 1. Создание результатов тестов на лекарственную чувствительность в рамках специфики по туберкулезу

POST api/MorbusTubStudyDrugResult

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubStudyResult\_id (N, O) – Ссылка на результаты исследований * MorbusTubStudyDrugResult\_setDT (D, О) – Дата проведения * TubDrug\_id (N, O) – Противотуберкулезный препарат (справочник dbo.TubDrug) * MorbusTubStudyDrugResult\_IsResult (N, H) – Результат теста (=2, если «Да», =1, если «Нет») |
| **Ответ** | **Успешный ответ идентификатор записи**  MorbusTubStudyDrugResult\_id (N, O) – Результаты тестов на лекарственную чувствительность |

* 1. Получение результатов тестов на лекарственную чувствительность в рамках специфики по туберкулезу

GET api/ MorbusTubStudyDrugResult

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubStudyDrugResult\_id (N, Н) – Результаты тестов на лекарственную чувствительность * MorbusTubStudyResult\_id (N, Н) – Результаты исследований |
| **Ответ** | Если не передан ни один параметр, то ошибка (не передан ни один из параметров).  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список с параметрами:   * MorbusTubStudyDrugResult\_id (N, Н) – Результаты тестов на лекарственную чувствительность * MorbusTubStudyResult\_id (N, Н) – Результаты исследований MorbusTubStudyDrugResult\_setDT (D, О) – Дата проведения * TubDrug\_id (N, O) – Противотуберкулезный препарат (справочник dbo.TubDrug) * MorbusTubStudyDrugResult\_IsResult (N, H) – Результат теста (=2, если «Да», =1, если «Нет») |

* 1. Изменение результатов тестов на лекарственную чувствительность в рамках специфики по туберкулезу

PUT api/ MorbusTubStudyDrugResult

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubStudyDrugResult\_id (N, О) – Результаты тестов на лекарственную чувствительность * MorbusTubStudyResult\_id (N, Н) – Результаты исследований MorbusTubStudyDrugResult\_setDT (D, Н) – Дата проведения * TubDrug\_id (N, Н) – Противотуберкулезный препарат (справочник dbo.TubDrug) * MorbusTubStudyDrugResult\_IsResult (N, H) – Результат теста (=2, если «Да», =1, если «Нет») |
| **Ответ** | Успешный ответ с ошибкой «0» |

* 1. Создание молекулярно–генетические методов в рамках специфики по туберкулезу

POST api/MorbusTubMolecular

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubStudyResult\_id (N, O) – Результаты исследований * MorbusTubMolecular\_IsResult (N, O) – Результат (1–Нет,2–Да) * MorbusTubMolecular\_setDT (D, О) – Дата проведения * MorbusTubMolecularType\_id (N, О) – Тест на лекарственную устойчивость (справочник dbo.MorbusTubMolecularType) |
| **Ответ** | **Успешный ответ идентификатор записи**   * MorbusTubMolecular\_id (N, O) – Молекулярно–генетические методы |

* 1. Получение молекулярно–генетические методов в рамках специфики по туберкулезу

GET api/MorbusTubMolecular

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubMolecular\_id (N, Н) – Молекулярно–генетические методы * MorbusTubStudyResult\_id (N, Н) – Результаты исследований |
| **Ответ** | Если не передан ни один параметр, то ошибка (не передан ни один из параметров).  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список с параметрами:   * MorbusTubMolecular\_id (N, O) – Молекулярно–генетические методы * MorbusTubStudyResult\_id (N, O) – Результаты исследований * MorbusTubMolecular\_IsResult (N, O) – Результат (1–Нет,2–Да) * MorbusTubMolecular\_setDT (D, О) – Дата проведения * MorbusTubMolecularType\_id (N, О) – тест на лекарственную устойчивость (справочник dbo.MorbusTubMolecularType) |

* 1. Изменение молекулярно–генетические методов в рамках специфики по туберкулезу

PUT api/MorbusTubMolecular

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubMolecular\_id (N, О) – Молекулярно–генетические методы * MorbusTubStudyResult\_id (N, Н) – Результаты исследований * MorbusTubMolecular\_IsResult (N, Н) – Результат (1–Нет,2–Да) * MorbusTubMolecular\_setDT (D, Н) – Дата проведения   MorbusTubMolecularType\_id (N, Н) – тест на лекарственную устойчивость (справочник dbo.MorbusTubMolecularType) |
| **Ответ** | Успешный ответ с ошибкой «0» |

* 1. Создание микроскопии в рамках специфики по туберкулезу

POST api/MorbusTubStudyMicrosResult

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubStudyResult\_id (N, O) – Результаты исследований * MorbusTubStudyMicrosResult\_EdResult (N, Н) – Число микобактерий * MorbusTubStudyMicrosResult\_NumLab (Т, Н) – Номер образца мокроты * MorbusTubStudyMicrosResult\_setDT (D, О) – Дата регистрации результата мокроты * TubMicrosResultType\_id (N, O) – Результат микроскопии (справочник dbo.TubMicrosResultType) |
| **Ответ** | **Успешный ответ идентификатор записи**  MorbusTubStudyMicrosResult\_id (N, O) – Результаты исследований микроскопии |

* 1. Получение микроскопии в рамках специфики по туберкулезу

GET api MorbusTubStudyMicrosResult

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubStudyMicrosResult\_id (N, O) – Результаты исследований микроскопии * MorbusTubStudyResult\_id (N, O) – Результаты исследований |
| **Ответ** | Если не передан ни один параметр, то ошибка (не передан ни один из параметров).  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список с параметрами:   * MorbusTubStudyResult\_id (N, O) – Результаты исследований * MorbusTubStudyMicrosResult\_id (N, O) – Результаты исследований микроскопии * MorbusTubStudyMicrosResult\_EdResult (N, Н) – Число микобактерий * MorbusTubStudyMicrosResult\_NumLab (Т, Н) – Номер образца мокроты * MorbusTubStudyMicrosResult\_setDT (D, О) – Дата регистрации результата мокроты * TubMicrosResultType\_id (N, O) – Результат микроскопии (справочник dbo.TubMicrosResultType) |

* 1. Изменение микроскопии в рамках специфики по туберкулезу

PUT api/MorbusTubStudyMicrosResult

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubStudyMicrosResult\_id (N, O) – Результаты исследований микроскопии * MorbusTubStudyResult\_id (N, Н) – Результаты исследований * MorbusTubStudyMicrosResult\_EdResult (N, Н) – Число микобактерий * MorbusTubStudyMicrosResult\_NumLab (Т, Н) – Номер образца мокроты * MorbusTubStudyMicrosResult\_setDT (D, Н) – Дата регистрации результата мокроты * TubMicrosResultType\_id (N, Н) – Результат микроскопии (справочник dbo.TubMicrosResultType |
| **Ответ** | Успешный ответ с ошибкой «0» |

* 1. Создание результатов посева в рамках специфики по туберкулезу

POST api/MorbusTubStudySeedResult

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubStudyResult\_id (N, O) – Результаты исследований * MorbusTubStudySeedResult\_setDT (D, О) – Дата проведения * TubSeedResultType\_id (N, O) – Результат посева (Справочник dbo.TubSeedResultType) |
| **Ответ** | **Успешный ответ идентификатор записи**   * MorbusTubStudySeedResult\_id (N, O) – Результаты исследований посева |

* 1. Получение результатов посева в рамках специфики по туберкулезу

GET api/MorbusTubStudySeedResult

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubStudySeedResult\_id (N, Н) – Результаты исследований посева * MorbusTubStudyResult\_id (N, Н) – Результаты исследований |
| **Ответ** | Если не передан ни один параметр, то ошибка (не передан ни один из параметров).  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список с параметрами:   * MorbusTubStudySeedResult\_id (N, О) – Результаты исследований посева * MorbusTub\_id (N, О) – Простое заболевание: Туберкулез * MorbusTubStudyResult\_id (N, О) – Результаты исследований MorbusTubStudySeedResult\_setDT (D, О) – Дата проведения * TubSeedResultType\_id (N, O) – Результат посева (Справочник dbo.TubSeedResultType) |

* 1. Изменение результатов посева в рамках специфики по туберкулезу

PUT api/MorbusTubStudySeedResult

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubStudySeedResult\_id (N, О) – Результаты исследований посева * MorbusTubStudyResult\_id (N, Н) – Результаты исследований MorbusTubStudySeedResult\_setDT (D, Н) – Дата проведения * TubSeedResultType\_id (N, Н) – Результат посева (Справочник dbo.TubSeedResultType) |
| **Ответ** | Успешный ответ с ошибкой «0» |

* 1. Создание результатов гистологии в рамках специфики по туберкулезу

POST api/MorbusTubStudyHistolResult

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubStudyResult\_id (N, O) – Результаты исследований * MorbusTubStudyHistolResult\_setDT (D, O) – Дата проведения * TubDiagnosticMaterialType\_id (N, O) – Диагностический материал (справочник dbo.TubDiagnosticMaterialType) * TubHistolResultType\_id (N, O) – Результат гистологии (справочник dbo.TubHistolResultType) |
| **Ответ** | **Успешный ответ идентификатор записи**   * MorbusTubStudyHistolResult\_id (N, O) – Идентификатор результата исследований гистологии |

* 1. Получение результатов гистологии в рамках специфики по туберкулезу

GET api/MorbusTubStudyHistolResult

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubStudyHistolResult\_id (N, Н) – Идентификатор результата исследований гистологии * MorbusTubStudyResult\_id (N, Н) – Результаты исследований |
| **Ответ** | Если не передан ни один параметр, то ошибка (не передан ни один из параметров).  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список с параметрами:   * MorbusTubStudyHistolResult\_id (N, O) – Идентификатор результата исследований гистологии * MorbusTubStudyResult\_id (N, O) – Результаты исследований * MorbusTubStudyHistolResult\_setDT (D, O) – Дата проведения * TubDiagnosticMaterialType\_id (N, O) – Диагностический материал (справочник dbo.TubDiagnosticMaterialType) * TubHistolResultType\_id (N, O) – Результат гистологии (справочник dbo.TubHistolResultType) |

* 1. Изменение результатов гистологии в рамках специфики по туберкулезу

PUT api/MorbusTubStudyHistolResult

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubStudyHistolResult\_id (N, O) – Идентификатор результата исследований гистологии * MorbusTubStudyResult\_id (N, Н) – Результаты исследований * MorbusTubStudyHistolResult\_setDT (D, Н) – Дата проведения * TubDiagnosticMaterialType\_id (N, Н) – Диагностический материал (справочник dbo.TubDiagnosticMaterialType) * TubHistolResultType\_id (N, Н) – Результат гистологии (справочник dbo.TubHistolResultType) |
| **Ответ** | Успешный ответ с ошибкой «0» |

* 1. Создание результатов рентгена в рамках специфики по туберкулезу

POST api/MorbusTubStudyXrayResult

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubStudyResult\_id (N, O) – Результаты исследований * MorbusTubStudyXrayResult\_setDT (D, О) – Дата рентгена * TubXrayResultType\_id (N, O) – Результат рентгена (справочник dbo.TubXrayResultType) * MorbusTubStudyXrayResult\_Comment (Т, Н) – Примечание |
| **Ответ** | **Успешный ответ идентификатор записи**   * MorbusTubStudyXrayResult\_id (N, O) – Идентификатор результата исследований рентгена |

* 1. Получение результатов рентгена в рамках специфики по туберкулезу

GET api/MorbusTubStudyXrayResult

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubStudyXrayResult\_id (N, Н) – Идентификатор результата исследований рентгена * MorbusTubStudyResult\_id (N, Н) – Результаты исследований |
| **Ответ** | Если не передан ни один параметр, то ошибка (не передан ни один из параметров).  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список с параметрами:   * MorbusTubStudyXrayResult\_id (N, O) – Идентификатор результата исследований рентгена * MorbusTubStudyResult\_id (N, O) – Результаты исследований * MorbusTubStudyXrayResult\_setDT (D, О) – Дата рентгена * TubXrayResultType\_id (N, O) – Результат рентгена (справочник dbo.TubXrayResultType) * MorbusTubStudyXrayResult\_Comment (Т, Н) – Примечание |

* 1. Изменение результатов рентгена в рамках специфики по туберкулезу

PUT api/MorbusTubStudyXrayResult

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusTubStudyXrayResult\_id (N, O) – Идентификатор результата исследований рентгена * MorbusTubStudyResult\_id (N, Н) – Результаты исследований * MorbusTubStudyXrayResult\_setDT (D, Н) – Дата рентгена * TubXrayResultType\_id (N, Н) – Результат рентгена (справочник dbo.TubXrayResultType) * MorbusTubStudyXrayResult\_Comment (Т, Н) – Примечание |
| **Ответ** | Успешный ответ с ошибкой «0» |

* 1. Создание извещения об онкобольном

POST api/EvnOnkoNotify

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N,O) – ссылка на человека в Person * Evn\_pid (N,O) – идентификатор родительского события * Lpu\_sid (N, О) – Направить извещение * MedPersonal\_id (N, О) – Врач, заполнивший извещение * EvnOnkoNotify\_setDT (D,O) – Дата создания извещения * EvnOnkoNotify\_setDiagDT (D, O) – дата установки диагноза * EvnOnkoNotify\_IsDiagConfCito (N, Н) – метод подтверждения диагноза: Цитологический (Да – 2, Нет – 1) * EvnOnkoNotify\_IsDiagConfClinic (N, Н) – метод подтверждения диагноза: Только клинический (Да – 2, Нет – 1) * EvnOnkoNotify\_IsDiagConfExplo (N, Н) – метод подтверждения диагноза: Эксплоротивная операция (Да – 2, Нет – 1) * EvnOnkoNotify\_IsDiagConfLab (N, Н) – метод подтверждения диагноза: Лабораторно-инструментальный (Да – 2, Нет – 1) * EvnOnkoNotify\_IsDiagConfMorfo (N, Н) – метод подтверждения диагноза: Морфологический (Да – 2, Нет – 1) * EvnOnkoNotify\_IsDiagConfUnknown (N, Н) – метод подтверждения диагноза: Неизвестен (Да – 2, Нет – 1) * OnkoT\_id (N, О) – T; * OnkoN\_id (N, О) – N; * OnkoM\_id (N, О) – M; * TumorStage\_id (N, О) – Стадия опухолевого процесса (справочник dbo.TumorStage);   --Локализация отдаленных метастазов (хотя бы один из параметров должен быть заполнен):   * EvnOnkoNotify\_IsTumorDepoBones (N, У) – локализация отдаленных метастазов: Кости (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoBrain (N, У) – локализация отдаленных метастазов: Головной мозг (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoKidney (N, У) – локализация отдаленных метастазов: Почки (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoLiver (N, У) – локализация отдаленных метастазов: Печень (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoLungs (N, У) – локализация отдаленных метастазов: Легкие и/или плевра (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoLympha (N, У) – локализация отдаленных метастазов: Отдаленные лимфатические узлы (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoMarrow (N, У) – локализация отдаленных метастазов: Костный мозг (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoMulti (N, У) – локализация отдаленных метастазов: Множественные (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoOther (N, У) – локализация отдаленных метастазов: Другие органы (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoOvary (N, У) – локализация отдаленных метастазов: Яичники (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoPerito (N, У) – локализация отдаленных метастазов: Брюшина (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoSkin (N, У) – локализация отдаленных метастазов: Кожа (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoUnknown (N, У) – локализация отдаленных метастазов: Неизвестна (Да – 2, Нет – 1) |
| **Ответ** | **Успешный ответ идентификатор записи**   * EvnOnkoNotify\_id (N, O) – Идентификатор извещения   При создании извещения проводится проверка:  Если у пациента присутствует извещение об онкологии, то ошибка «Извещение на пациента отправлено». |

* 1. Получение извещения об онкобольном

GET api/EvnOnkoNotify

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnOnkoNotify\_id (N, Н) – Идентификатор извещения * Person\_id (N,H) – ссылка на человека в Person |
| **Ответ** | Если не передан ни один параметр, то ошибка (не передан ни один из параметров).  Если не найдено ни одной записи, то ответ «0».  Если хотя бы одна запись найдена, то успешный ответ – список с параметрами:   * EvnOnkoNotify\_id (N, О) – Идентификатор извещения * Person\_id (N,O) – ссылка на человека в Person * Evn\_pid (N,O) – идентификатор родительского события * Lpu\_sid (N, О) – Направить извещение * MedPersonal\_id (N, О) – Врач, заполнивший извещение * EvnOnkoNotify\_setDT (D,O) – Дата создания извещения * EvnOnkoNotify\_setDiagDT (D, O) – дата установки диагноза * EvnOnkoNotify\_IsDiagConfCito (N, Н) – метод подтверждения диагноза: Цитологический (Да – 2, Нет – 1) * EvnOnkoNotify\_IsDiagConfClinic (N, Н) – метод подтверждения диагноза: Только клинический (Да – 2, Нет – 1) * EvnOnkoNotify\_IsDiagConfExplo (N, Н) – метод подтверждения диагноза: Эксплоротивная операция (Да – 2, Нет – 1) * EvnOnkoNotify\_IsDiagConfLab (N, Н) – метод подтверждения диагноза: Лабораторно-инструментальный (Да – 2, Нет – 1) * EvnOnkoNotify\_IsDiagConfMorfo (N, Н) – метод подтверждения диагноза: Морфологический (Да – 2, Нет – 1) * EvnOnkoNotify\_IsDiagConfUnknown (N, Н) – метод подтверждения диагноза: Неизвестен (Да – 2, Нет – 1) * OnkoT\_id (N, О) – T; * OnkoN\_id (N, О) – N; * OnkoM\_id (N, О) – M; * TumorStage\_id (N, О) – Стадия опухолевого процесса (справочник dbo.TumorStage); * EvnOnkoNotify\_IsTumorDepoBones (N, Н) – локализация отдаленных метастазов: Кости (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoBrain (N, Н) – локализация отдаленных метастазов: Головной мозг (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoKidney (N, Н) – локализация отдаленных метастазов: Почки (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoLiver (N, Н) – локализация отдаленных метастазов: Печень (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoLungs (N, Н) – локализация отдаленных метастазов: Легкие и/или плевра (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoLympha (N, Н) – локализация отдаленных метастазов: Отдаленные лимфатические узлы (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoMarrow (N, Н) – локализация отдаленных метастазов: Костный мозг (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoMulti (N, Н) – локализация отдаленных метастазов: Множественные (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoOther (N, Н) – локализация отдаленных метастазов: Другие органы (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoOvary (N, Н) – локализация отдаленных метастазов: Яичники (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoPerito (N, Н) – локализация отдаленных метастазов: Брюшина (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoSkin (N, Н) – локализация отдаленных метастазов: Кожа (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoUnknown (N, Н) – локализация отдаленных метастазов: Неизвестна (Да – 2, Нет – 1)   --Обязательные параметры при отказе включения в регистр:   * Lpu\_niid (N, У) – Лпу, не включившее в регистр * MedPersonal\_niid (N, У) – Врач, не включивший в регистр * PersonRegisterFailIncludeCause\_id (N, У) – Причина невключения в регистр (справочник dbo.PersonRegisterFailIncludeCause) * EvnOnkoNotify\_Comment (N, Н) – Комментариий |

* 1. Изменение извещения об онкобольном

PUT api/EvnOnkoNotify

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnOnkoNotify\_id (N, O) – Идентификатор извещения * Person\_id (N,Н) – ссылка на человека в Person * Evn\_pid (N,Н) – идентификатор родительского события * Lpu\_sid (N, Н) – Направить извещение * MedPersonal\_id (N, Н) – Врач, заполнивший извещение * EvnOnkoNotify\_setDT (D,Н) – Дата создания извещения * EvnOnkoNotify\_setDiagDT (D, Н) – дата установки диагноза * EvnOnkoNotify\_IsDiagConfCito (N, Н) – метод подтверждения диагноза: Цитологический (Да – 2, Нет – 1) * EvnOnkoNotify\_IsDiagConfClinic (N, Н) – метод подтверждения диагноза: Только клинический (Да – 2, Нет – 1) * EvnOnkoNotify\_IsDiagConfExplo (N, Н) – метод подтверждения диагноза: Эксплоротивная операция (Да – 2, Нет – 1) * EvnOnkoNotify\_IsDiagConfLab (N, Н) – метод подтверждения диагноза: Лабораторно-инструментальный (Да – 2, Нет – 1) * EvnOnkoNotify\_IsDiagConfMorfo (N, Н) – метод подтверждения диагноза: Морфологический (Да – 2, Нет – 1) * EvnOnkoNotify\_IsDiagConfUnknown (N, Н) – метод подтверждения диагноза: Неизвестен (Да – 2, Нет – 1) * OnkoT\_id (N, Н) – T; * OnkoN\_id (N, Н) – N; * OnkoM\_id (N, Н) – M; * TumorStage\_id (N, Н) – Стадия опухолевого процесса (справочник dbo.TumorStage); * EvnOnkoNotify\_IsTumorDepoBones (N, Н) – локализация отдаленных метастазов: Кости (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoBrain (N, Н) – локализация отдаленных метастазов: Головной мозг (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoKidney (N, Н) – локализация отдаленных метастазов: Почки (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoLiver (N, Н) – локализация отдаленных метастазов: Печень (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoLungs (N, Н) – локализация отдаленных метастазов: Легкие и/или плевра (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoLympha (N, Н) – локализация отдаленных метастазов: Отдаленные лимфатические узлы (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoMarrow (N, Н) – локализация отдаленных метастазов: Костный мозг (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoMulti (N, Н) – локализация отдаленных метастазов: Множественные (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoOther (N, Н) – локализация отдаленных метастазов: Другие органы (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoOvary (N, Н) – локализация отдаленных метастазов: Яичники (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoPerito (N, Н) – локализация отдаленных метастазов: Брюшина (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoSkin (N, Н) – локализация отдаленных метастазов: Кожа (Да – 2, Нет – 1) * EvnOnkoNotify\_IsTumorDepoUnknown (N, Н) – локализация отдаленных метастазов: Неизвестна (Да – 2, Нет – 1)   --Обязательные параметры при отказе включения в регистр:   * Lpu\_niid (N, У) – Лпу, не включившее в регистр * MedPersonal\_niid (N, У) – Врач, не включивший в регистр * PersonRegisterFailIncludeCause\_id (N, У) – Причина невключения в регистр (справочник dbo.PersonRegisterFailIncludeCause) * EvnOnkoNotify\_Comment (N, Н) – Комментариий |
| **Ответ** | Успешный ответ с ошибкой «0» |

* 1. Создание данных по специфике онкологии

**POST api/MorbusOnko**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, Н) – Ид человека; * PersonRegister\_id (N, О) – идентификатор записи регистра * Evn\_pid (N, Н) – Ид события-родителя (сохраняется связь в таблице dbo.MorbusBase); передается ссылка на родительский случай: движение/посещение * OnkoTreatment\_id (N, Н) – Повод обращения (справочник dbo.OnkoTreatment) * MorbusOnko\_firstSignDT (D, Н) – Дата появления первых признаков заболевания; * MorbusOnko\_firstVizitDT (D, Н) – Дата первого обращения; * Lpu\_foid (N, Н) – В какое МО; * MorbusOnko\_setDiagDT (D, Н) – Дата установления диагноза; * MorbusOnko\_NumCard (Т, Н) – Регистрационный номер * MorbusOnko\_NumTumor (N, Н) – Порядковый номер данной опухоли. * TumorPrimaryMultipleType\_id (N, Н) – Первично-множественная опухоль (справочник dbo.TumorPrimaryMultipleType) * MorbusOnko\_IsMainTumor (N, Н) – Признак основной опухоли (2-Да, 1-Нет); * Diag\_id (N, О) – Топография (локализация) опухоли; * OnkoLesionSide\_id (N, Н) – Сторона поражения (справочник dbo.OnkoLesionSide); * OnkoDiagConfType\_id (N, Н) – Метод подтверждения диагноза (справочник dbo.OnkoDiagConfType); * MorbusOnko\_takeDT (D, Н) – Дата взятия материала * DiagAttribType\_id (N, Н) – Тип диагностического показателя (справочник dbo.DiagAttribType) * DiagResult\_id (N, Н) – Результат диагностики (справочник dbo.DiagResult) * DiagAttribDict\_id (N, Н) – Диагностический показатель * Морфологический тип опухоли (справочник dbo.DiagAttribDict) * OnkoDiag\_mid (N, Н) – Морфологический тип опухоли. (Гистология опухоли) (справочник dbo.OnkoDiag) * MorbusOnko\_NumHisto (Т, Н) – Номер гистологического исследования; * OnkoT\_id (N, О) – T; * OnkoN\_id (N, О) – N; * OnkoM\_id (N, О) – M; * TumorStage\_id (N, О) – Стадия опухолевого процесса (справочник dbo.TumorStage); * MorbusOnko\_IsTumorDepoUnknown (N, Н) – Локализация отдельных метастазов: Неизвестна (2-Да, 1-Нет); * MorbusOnko\_IsTumorDepoLympha (N, Н) – Локализация отдельных метастазов: Отдаленные лимфотические узлы(2-Да, 1-Нет); * MorbusOnko\_IsTumorDepoBones (N, Н) – Локализация отдельных метастазов: Кости (2-Да, 1-Нет); * MorbusOnko\_IsTumorDepoLiver (N, Н) – Локализация отдельных метастазов: Печень (2-Да, 1-Нет); * MorbusOnko\_IsTumorDepoLungs (N, Н) – Локализация отдельных метастазов: Легкие и/или плевра (2-Да, 1-Нет); * MorbusOnko\_IsTumorDepoBrain (N, Н) – Локализация отдельных метастазов: Головной мозг (2-Да, 1-Нет); * MorbusOnko\_IsTumorDepoSkin (N, Н) – Локализация отдельных метастазов: Кожа (2-Да, 1-Нет); * MorbusOnko\_IsTumorDepoKidney (N, Н) – Локализация отдельных метастазов: Почки (2-Да, 1-Нет); * MorbusOnko\_IsTumorDepoOvary (N, Н) – Локализация отдельных метастазов: Яичники (2-Да, 1-Нет); * MorbusOnko\_IsTumorDepoPerito (N, Н) – Локализация отдельных метастазов: Брюшина (2-Да, 1-Нет); * MorbusOnko\_IsTumorDepoMarrow (N, Н) – Локализация отдельных метастазов: Костный мозг (2-Да, 1-Нет); * MorbusOnko\_IsTumorDepoOther (N, Н) – Локализация отдельных метастазов: Другие органы (2-Да, 1-Нет); * MorbusOnko\_IsTumorDepoMulti (N, Н) – Локализация отдельных метастазов: Множественные (2-Да, 1-Нет); * OnkoPostType\_id (N, Н) – Выявлен врачом (справочник dbo.OnkoPostType); * TumorCircumIdentType\_id (N, Н) – Обстоятельства выявления опухоли (справочник dbo.TumorCircumIdentType); * OnkoLateDiagCause\_id (N, Н) – Причины поздней диагностики (справочник dbo.OnkoLateDiagCause); * MorbusOnkoBase\_deadDT (D, Н) – Дата смерти * Diag\_id (N, Н) – Причина смерти * AutopsyPerformType\_id (N, Н) – Аутопсия (справочник dbo.AutopsyPerformType); * TumorAutopsyResultType\_id (N, Н) – Результат аутопсии применительно к данной опухоли (справочник dbo. TumorAutopsyResultType); * OnkoStatusYearEndType\_id (N, Н) – Клиническая группа (справочник dbo.OnkoStatusYearEndType); * OnkoInvalidType\_id (N, Н) – Инвалидность по основному заболеванию (dbo.OnkoInvalidType). * MorbusOnko\_MorfoDiag (N, Н) – Подробный морфологический диагноз * TumorPrimaryTreatType\_id (N, Н) – Проведенное лечение первичной опухоли; * TumorRadicalTreatIncomplType\_id (N, Н) – Причины незавершенности радикального лечения;   ---Специальное лечение   * MorbusOnko\_specSetDT (D, Н) – Дата начала специального лечения; * MorbusOnko\_specDisDT (D, Н) – Дата окончания специального лечения; * OnkoCombiTreatType\_id (N, Н) – Сочетание видов лечения (справочник dbo.OnkoCombiTreatType); * OnkoLateComplTreatType\_id (N, МН) – Позднее осложнение специального лечения (справочник dbo.OnkoLateComplTreatType); * OnkoTumorStatusType\_id (N, Н) – Состояние опухолевого процесса (справочник dbo.OnkoTumorStatusType); * MorbusOnko\_Deleted (N, Н) – Признак удаления; |
| **Ответ** | **Успешный ответ идентификатор записи**   * MorbusOnko\_id (N, О) – Ид заболевания по онкологии; |

* 1. Изменение данных по специфике онкологии

**PUT api/MorbusOnko**

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusOnko\_id (N, О) – Ид заболевания по онкологии; * PersonRegister\_id (N, Н) – идентификатор записи регистра * Evn\_pid (N, Н) – Ид события-родителя (связь в таблице dbo.MorbusBase) * Morbus\_id (N, О) – Ид заболевания; * OnkoTreatment\_id (N, Н) – Повод обращения (справочник dbo.OnkoTreatment) * MorbusOnko\_firstSignDT (D, Н) – Дата появления первых признаков заболевания; * MorbusOnko\_firstVizitDT (D, Н) – Дата первого обращения; * Lpu\_foid (N, Н) – В какое МО; * MorbusOnko\_setDiagDT (D, Н) – Дата установления диагноза; * MorbusOnko\_NumCard (Т, Н) – Регистрационный номер * MorbusOnko\_NumTumor (N, Н) – Порядковый номер данной опухоли. * TumorPrimaryMultipleType\_id (N, Н) – Первично-множественная опухоль (справочник dbo.TumorPrimaryMultipleType) * MorbusOnko\_IsMainTumor (N, Н) – Признак основной опухоли (2-Да, 1-Нет); * Diag\_id (N, Н) – Топография (локализация) опухоли; * OnkoLesionSide\_id (N, Н) – Сторона поражения (справочник dbo.OnkoLesionSide); * OnkoDiagConfType\_id (N, Н) – Метод подтверждения диагноза (справочник dbo.OnkoDiagConfType); * MorbusOnko\_takeDT (D, Н) – Дата взятия материала * DiagAttribType\_id (N, Н) – Тип диагностического показателя (справочник dbo.DiagAttribType) * DiagResult\_id (N, Н) – Результат диагностики (справочник dbo.DiagResult) * DiagAttribDict\_id (N, Н) – Диагностический показатель * Морфологический тип опухоли (справочник dbo.DiagAttribDict) * OnkoDiag\_mid (N, Н) – Морфологический тип опухоли. (Гистология опухоли) (справочник dbo.OnkoDiag) * MorbusOnko\_NumHisto (Т, Н) – Номер гистологического исследования; * OnkoT\_id (N, Н) – T; * OnkoN\_id (N, Н) – N; * OnkoM\_id (N, Н) – M; * TumorStage\_id (N, Н) – Стадия опухолевого процесса (справочник dbo.TumorStage); * MorbusOnko\_IsTumorDepoUnknown (N, Н) – Локализация отдельных метастазов: Неизвестна (2-Да, 1-Нет); * MorbusOnko\_IsTumorDepoLympha (N, Н) – Локализация отдельных метастазов: Отдаленные лимфотические узлы(2-Да, 1-Нет); * MorbusOnko\_IsTumorDepoBones (N, Н) – Локализация отдельных метастазов: Кости (2-Да, 1-Нет); * MorbusOnko\_IsTumorDepoLiver (N, Н) – Локализация отдельных метастазов: Печень (2-Да, 1-Нет); * MorbusOnko\_IsTumorDepoLungs (N, Н) – Локализация отдельных метастазов: Легкие и/или плевра (2-Да, 1-Нет); * MorbusOnko\_IsTumorDepoBrain (N, Н) – Локализация отдельных метастазов: Головной мозг (2-Да, 1-Нет); * MorbusOnko\_IsTumorDepoSkin (N, Н) – Локализация отдельных метастазов: Кожа (2-Да, 1-Нет); * MorbusOnko\_IsTumorDepoKidney (N, Н) – Локализация отдельных метастазов: Почки (2-Да, 1-Нет); * MorbusOnko\_IsTumorDepoOvary (N, Н) – Локализация отдельных метастазов: Яичники (2-Да, 1-Нет); * MorbusOnko\_IsTumorDepoPerito (N, Н) – Локализация отдельных метастазов: Брюшина (2-Да, 1-Нет); * MorbusOnko\_IsTumorDepoMarrow (N, Н) – Локализация отдельных метастазов: Костный мозг (2-Да, 1-Нет); * MorbusOnko\_IsTumorDepoOther (N, Н) – Локализация отдельных метастазов: Другие органы (2-Да, 1-Нет); * MorbusOnko\_IsTumorDepoMulti (N, Н) – Локализация отдельных метастазов: Множественные (2-Да, 1-Нет); * OnkoPostType\_id (N, Н) – Выявлен врачом (справочник dbo.OnkoPostType); * TumorCircumIdentType\_id (N, Н) – Обстоятельства выявления опухоли (справочник dbo.TumorCircumIdentType); * OnkoLateDiagCause\_id (N, Н) – Причины поздней диагностики (справочник dbo.OnkoLateDiagCause); * MorbusOnkoBase\_deadDT (D, Н) – Дата смерти * Diag\_id (N, Н) – Причина смерти * AutopsyPerformType\_id (N, Н) – Аутопсия (справочник dbo.AutopsyPerformType); * TumorAutopsyResultType\_id (N, Н) – Результат аутопсии применительно к данной опухоли (справочник dbo. TumorAutopsyResultType); * OnkoStatusYearEndType\_id (N, Н) – Клиническая группа (справочник dbo.OnkoStatusYearEndType); * OnkoInvalidType\_id (N, Н) – Инвалидность по основному заболеванию (dbo.OnkoInvalidType). * MorbusOnko\_MorfoDiag (N, Н) – Подробный морфологический диагноз * TumorPrimaryTreatType\_id (N, Н) – Проведенное лечение первичной опухоли; * TumorRadicalTreatIncomplType\_id (N, Н) – Причины незавершенности радикального лечения;   ---Специальное лечение   * MorbusOnko\_specSetDT (D, Н) – Дата начала специального лечения; * MorbusOnko\_specDisDT (D, Н) – Дата окончания специального лечения; * OnkoCombiTreatType\_id (N, Н) – Сочетание видов лечения (справочник dbo.OnkoCombiTreatType); * OnkoLateComplTreatType\_id (N, МН) – Позднее осложнение специального лечения (справочник dbo.OnkoLateComplTreatType); * OnkoTumorStatusType\_id (N, Н) – Состояние опухолевого процесса (справочник dbo.OnkoTumorStatusType); * MorbusOnko\_Deleted (N, Н) – Признак удаления |
| **Ответ** | Успешный ответ с ошибкой «0» |

* 1. Получение данных по специфике онкологии

**GET api/MorbusOnko**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, Н) – Ид человека; * Evn\_pid (N, Н) – Ид события-родителя (связь в таблице dbo.MorbusBase) |
| **Ответ** | Проверка: хотя бы один из параметров Person\_id, Evn\_pid должен быть заполнен. Иначе ошибка.  Успешный ответ:   * Person\_id – Ид человека; * Evn\_pid – Ид события-родителя (связь в таблице dbo.MorbusBase); * MorbusOnko\_id – Ид заболевания по онкологии; * Morbus\_id – Ид заболевания; * MorbusOnko\_firstSignDT – Дата появления первых признаков заболевания; * MorbusOnko\_firstVizitDT – Дата первого обращения; * Lpu\_foid – В какое МО; * Diag\_id – Топография (локализация) опухоли; * OnkoLesionSide\_id – Сторона поражения; * MorbusOnko\_MorfoDiag – Подробный морфологический диагноз; * MorbusOnko\_NumHisto – Номер гистологического исследования; * OnkoT\_id – T; * OnkoN\_id – N; * OnkoM\_id – M; * TumorStage\_id – Стадия опухолевого процесса; * MorbusOnko\_IsTumorDepoUnknown – Локализация отдельных метастазов: Неизвестна; * MorbusOnko\_IsTumorDepoLympha – Локализация отдельных метастазов: Отдаленные лимфотические узлы; * MorbusOnko\_IsTumorDepoBones – Локализация отдельных метастазов: Кости; * MorbusOnko\_IsTumorDepoLiver – Локализация отдельных метастазов: Печень; * MorbusOnko\_IsTumorDepoLungs – Локализация отдельных метастазов: Легкие и/или плевра; * MorbusOnko\_IsTumorDepoBrain – Локализация отдельных метастазов: Головной мозг; * MorbusOnko\_IsTumorDepoSkin – Локализация отдельных метастазов: Кожа; * MorbusOnko\_IsTumorDepoKidney – Локализация отдельных метастазов: Почки; * MorbusOnko\_IsTumorDepoOvary – Локализация отдельных метастазов: Яичники; * MorbusOnko\_IsTumorDepoPerito – Локализация отдельных метастазов: Брюшина; * MorbusOnko\_IsTumorDepoMarrow – Локализация отдельных метастазов: Костный мозг; * MorbusOnko\_IsTumorDepoOther – Локализация отдельных метастазов: Другие органы; * MorbusOnko\_IsTumorDepoMulti – Локализация отдельных метастазов: Множественные; * TumorCircumIdentType\_id – Обстоятельства выявления опухоли; * OnkoLateDiagCause\_id – Причины поздней дианостики; * TumorAutopsyResultType\_id – Результат аутопсии применительно к данной опухоли; * TumorPrimaryTreatType\_id – Проведенное лечение первичной опухоли; * TumorRadicalTreatIncomplType\_id – Причины незавершенности радикального лечения; * MorbusOnko\_specSetDT – Дата начала специального лечения; * MorbusOnko\_specDisDT – Дата окончания специального лечения; * MorbusOnko\_Deleted – Признак удаления; * MorbusOnko\_IsMainTumor – Признак основной опухоли; * MorbusOnko\_setDiagDT – Дата установления диагноза; * OnkoDiag\_mid – Морфологический тип опухоли; * OnkoTumorStatusType\_id – Состояние опухолевого процесса; * OnkoDiagConfType\_id – Метод подтверждения диагноза; * OnkoPostType\_id – Выявлен врачом; * OnkoLateComplTreatType\_id – Позднее осложнение специального лечения; * OnkoCombiTreatType\_id – Сочетание видов лечения; * MorbusOnko\_NumTumor – Порядковый номер данной опухоли. |

* 1. Создание данных по химиотерапевтическому лечению в рамках специфики онкологии

**POST api/EvnUslugaOnkoChem**

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusOnko\_id (N, O) – Ид заболевания по онкологии. * UslugaComplex\_id (N, Н) – Ид услуги (справочник dbo.UslugaComplex) * Evn\_setDT (DT, O) – Дата начала выполнения услуги; * Evn\_disDT (DT, H) – Дата окончания выполнения услуги; * OnkoUslugaChemKindType\_id (N, O) – Вид проведенного химиотерапевтического лечения (справочник dbo.OnkoUslugaChemKindType); * OnkoUslugaChemFocusType\_id (N, O) – Преимущественная направленность химиотерапии (справочник dbo.OnkoUslugaChemFocusType); * Lpu\_uid (N, Н) – Место выполнения * TreatmentConditionsType\_id (N, Н) – Условия проведения (справочник dbo.TreatmentConditionsType); * AggType\_id (N, МН) – Осложнениt (справочник dbo.AggType); * OnkoTreatType\_id (N, Н) – Характер лечения (справочник dbo. OnkoTreatType); * DrugTherapyLineType\_id (N, O) – Линия лекарственной терапии (справочник dbo.DrugTherapyLineType) * DrugTherapyLoopType\_id (N, O) – Цикл лекарственной терапии (справочник dbo.DrugTherapyLoopType) * EvnUslugaOnkoChem\_Scheme (T, Н) – Схема химиотерапии * OnkoUslugaChemStageType\_id (N, Н) – Этапы лечения по химиотерапии (dbo.OnkoUslugaChemStageType) |
| **Ответ** | **Успешный ответ идентификатор записи**   * EvnUslugaOnkoChem\_id – Ид услуги по химиотерапевтическому лечению; |

* 1. Изменение данных по химиотерапевтическому лечению в рамках специфики онкологии

**PUT api/EvnUslugaOnkoChem**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnUslugaOnkoChem\_id (N, О) – Ид услуги по химиотерапевтическому лечению; * MorbusOnko\_id (N, Н) – Ид заболевания по онкологии. * UslugaComplex\_id (N, Н) – Ид услуги (справочник dbo.UslugaComplex) * Evn\_setDT (DT, О) – Дата начала выполнения услуги; * Evn\_disDT (DT, H) – Дата окончания выполнения услуги; * OnkoUslugaChemKindType\_id (N, Н) – Вид проведенного химиотерапевтического лечения (справочник dbo.OnkoUslugaChemKindType); * OnkoUslugaChemFocusType\_id (N, Н) – Преимущественная направленность химиотерапии (справочник dbo.OnkoUslugaChemFocusType); * Lpu\_uid (N, Н) – Место выполнения * TreatmentConditionsType\_id (N, Н) – Условия проведения (справочник dbo.TreatmentConditionsType); * AggType\_id (N, МН) – Осложнениt (справочник dbo.AggType); * OnkoTreatType\_id (N, Н) – Характер лечения (справочник dbo. OnkoTreatType); * DrugTherapyLineType\_id (N, Н) – Линия лекарственной терапии (справочник dbo.DrugTherapyLineType) * DrugTherapyLoopType\_id (N, Н) – Цикл лекарственной терапии (справочник dbo.DrugTherapyLoopType) * EvnUslugaOnkoChem\_Scheme (T, Н) – Схема химиотерапии * OnkoUslugaChemStageType\_id (N, Н) – Этапы лечения по химиотерапии (dbo.OnkoUslugaChemStageType) |
| **Ответ** | * Успешный ответ с ошибкой «0» |
| **Пример** |  |

* 1. Получение данных по химиотерапевтическому лечению в рамках специфики онкологии

**GET api/EvnUslugaOnkoChem**

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusOnko\_id (N, Н) – Ид заболевания по онкологии. |
| **Ответ** | Успешный ответ:   * MorbusOnko\_id – Ид заболевания по онкологии. * EvnUslugaOnkoChem\_id – Ид услуги п химиотерапевтическому лечению; * EvnUsluga\_id – Ид услуги; * Evn\_setDT – Дата начала выполнения услуги; * Evn\_disDT – Дата окончания выполнения услуги; * OnkoUslugaChemKindType\_id – Вид проведенного химиотерапевтического лечения; * OnkoUslugaChemFocusType\_id – Преимущественная направленность химиотерапии; * TreatmentConditionsType\_id – Условия проведения; * AggType\_id – Тип осложнения; * OnkoTreatType\_id – Характер лечения; * EvnUslugaOnkoChem\_Scheme – Схема химиотерапии; * OnkoUslugaChemStageType\_id. |

* 1. Создание данных по лучевому лечению в рамках специфики онкологии

**POST api/EvnUslugaOnkoBeam**

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusOnko\_id (N, O) – Ид заболевания по онкологии. * UslugaComplex\_id (N, Н) – Ид услуги (справочник dbo.UslugaComplex) * EvnUslugaOnkoBeam\_setDT (DT, O) – Дата начала. * EvnUslugaOnkoBeam\_disDT (DT, H) – Дата окончания. * EvnUslugaOnkoBeam\_CountFractionRT (N, О) – количество фракций проведения лучевой терапии. * OnkoUslugaBeamIrradiationType\_id (N, O) – Способ облученияпри проведении лучевой терапии (справочник dbo.OnkoUslugaBeamIrradiationType); * OnkoUslugaBeamKindType\_id (N, O) – Вид лучевой терапии (справочник dbo.OnkoUslugaBeamKindType); * OnkoUslugaBeamMethodType\_id (N, O) – Метод лучевой терапии (справочник dbo.OnkoUslugaBeamMethodType); * OnkoUslugaBeamRadioModifType\_id (N, H) – Радиомодификаторы (справочник dbo.OnkoUslugaBeamRadioModifType); * OnkoUslugaBeamFocusType\_id (N, O) – Преимущественная направленность лучевой терапии (справочник dbo.OnkoUslugaBeamFocusType); * OnkoPlanType\_id (N, Н) – Вид планирования (справочник dbo.OnkoPlanType); * Lpu\_uid (N, Н) – Место выполнения * OnkoTreatType\_id (N, Н) – Характер лечения (справочник dbo. OnkoTreatType); * OnkoRadiotherapy\_id (N, O) – Тип лечения (справочник dbo. OnkoRadiotherapy); * TreatmentConditionsType\_id (N, Н) – Условия проведения (справочник dbo.TreatmentConditionsType); * AggType\_id (N, МН) – Осложнение (справочник dbo.AggType); * EvnUslugaOnkoBeam\_TotalDoseTumor (N, Н) – Суммарная доза облучения опухоли; * OnkoUslugaBeamUnitType\_id (N, Н) – Единица измерения суммарной дозы облучения зон регионарного метастазирования (справочник dbo.OnkoUslugaBeamUnitType; * EvnUslugaOnkoBeam\_TotalDoseRegZone (N, Н) – Суммарная доза облучения зон регионарного метастазирования; * OnkoUslugaBeamUnitType\_did (N, Н) – Единица измерения cуммарной дозы облучения зон регионарного метастазирования (справочник dbo.OnkoUslugaBeamUnitType); |
| **Ответ** | **Успешный ответ идентификатор записи**   * EvnUslugaOnkoBeam\_id (N, O) – Ид услуги по лучевому лечению; |
| **Пример** |  |

* 1. Изменения данных по лучевому лечению в рамках специфики онкологии

**PUT api/EvnUslugaOnkoBeam**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnUslugaOnkoBeam\_id (N, O) – Ид услуги по лучевому лечению * MorbusOnko\_id (N, Н) – Ид заболевания по онкологии. * UslugaComplex\_id (N, Н) – Ид услуги (справочник dbo.UslugaComplex) * EvnUslugaOnkoBeam\_setDT (DT, У) – Дата начала. * EvnUslugaOnkoBeam\_disDT (DT, H) – Дата окончания. * EvnUslugaOnkoBeam\_CountFractionRT (N, У) – количество фракций проведения лучевой терапии. * OnkoUslugaBeamIrradiationType\_id (N, Н) – Способ облучения при проведении лучевой терапии (справочник dbo.OnkoUslugaBeamIrradiationType); * OnkoUslugaBeamKindType\_id (N, Н) – Вид лучевой терапии (справочник dbo.OnkoUslugaBeamKindType); * OnkoUslugaBeamMethodType\_id (N, Н) – Метод лучевой терапии (справочник dbo.OnkoUslugaBeamMethodType); * OnkoUslugaBeamRadioModifType\_id (N, H) – Радиомодификаторы (справочник dbo.OnkoUslugaBeamRadioModifType); * OnkoUslugaBeamFocusType\_id (N, Н) – Преимущественная направленность лучевой терапии (справочник dbo.OnkoUslugaBeamFocusType); * OnkoPlanType\_id (N, Н) – Вид планирования (справочник dbo.OnkoPlanType); * Lpu\_uid (N, Н) – Место выполнения * OnkoTreatType\_id (N, Н) – Характер лечения (справочник dbo. OnkoTreatType); * OnkoRadiotherapy\_id (N, Н) – Тип лечения (справочник dbo. OnkoRadiotherapy); * TreatmentConditionsType\_id (N, Н) – Условия проведения (справочник dbo.TreatmentConditionsType); * AggType\_id (N, МН) – Осложнениt (справочник dbo.AggType); * EvnUslugaOnkoBeam\_TotalDoseTumor (N, Н) – Суммарная доза облучения опухоли; * OnkoUslugaBeamUnitType\_id (N, Н) – Единица измерения суммарной дозы облучения зон регионарного метастазирования (справочник dbo.OnkoUslugaBeamUnitType; * EvnUslugaOnkoBeam\_TotalDoseRegZone (N, Н) – Суммарная доза облучения зон регионарного метастазирования; * OnkoUslugaBeamUnitType\_did (N, Н) – Единица измерения cуммарной дозы облучения зон регионарного метастазирования (справочник dbo.OnkoUslugaBeamUnitType); |
| **Ответ** | * Успешный ответ с ошибкой «0» |
| **Пример** |  |

* 1. Получение данных по лучевому лечению в рамках специфики онкологии

**GET api/EvnUslugaOnkoBeam**

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusOnko\_id (N, Н) – Ид заболевания по онкологии. |
| **Ответ** | Успешный ответ:   * MorbusOnko\_id (N, Н) – Ид заболевания по онкологии. * EvnUslugaOnkoBeam\_id – Ид услуги по лучевому лечению; * EvnUsluga\_id – Ид услуги; * Evn\_setDT – Дата начала выполнения услуги; * Evn\_disDT – Дата окончания выполнения услуги; * EvnUslugaOnkoBeam\_CountFractionRT (N, О) – количество фракций проведения лучевой терапии. * OnkoUslugaBeamIrradiationType\_id – Способо облучения при проведении лучевой терапии; * OnkoUslugaBeamKindType\_id – Вид лучевой терапии; * OnkoUslugaBeamMethodType\_id – Метод лучевой терапии; * OnkoUslugaBeamRadioModifType\_id – Радиомодификаторы; * OnkoUslugaBeamFocusType\_id – Преимущественная направленность лучевой терапии; * EvnUslugaOnkoBeam\_TotalDoseTumor – Суммарная доза облучения опухоли; * OnkoUslugaBeamUnitType\_id – Единица измерения суммарной дозы облучения зон регионарного метастазирования; * EvnUslugaOnkoBeam\_TotalDoseRegZone – Суммарная доза облучения зон регионарного метастазирования; * OnkoUslugaBeamUnitType\_did – Единица измерения cуммарной дозы облучения зон регионарного метастазирования; * AggType\_id – Тип осложнения; * TreatmentConditionsType\_id – Условия проведения; * OnkoPlanType\_id – Вид планирования лучевого лечения; * OnkoTreatType\_id – Характер лечения; * EvnUslugaOnkoBeam\_TotalDoseLymph – Суммарная доза облучения на регионарные лимфоузлы. |
| **Пример** |  |

* 1. Создание данных по гормоноиммунотерапевтическому лечению в рамках специфики онкологии

**POST api/EvnUslugaOnkoGormun**

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusOnko\_id (N, O) – Ид заболевания по онкологии. * UslugaComplex\_id (N, O) – Ид услуги (справочник dbo.UslugaComplex) * EvnUslugaOnkoGormun\_setDT (DT, O) – Дата начала выполнения услуги; * EvnUslugaOnkoGormun\_disDT (DT, Н) – Дата окончания выполнения услуги; * Lpu\_uid (N, Н) – Место выполнения * OnkoUslugaGormunFocusType\_id (N, O) – Преимущественная направленность гормоноиммунотерапии (справочник dbo.OnkoUslugaGormunFocusType); * OnkoTreatType\_id (N, Н) – Характер лечения (справочник dbo.OnkoTreatType) * TreatmentConditionsType\_id – Условия проведения (справочник dbo.TreatmentConditionsType); * AggType\_id (N, Н) – Тип осложнения (справочник dbo.AggType); * EvnUslugaOnkoGormun\_IsBeam (N, Н) – Признак «Лучевая» 0 – нет, 1 – да; * EvnUslugaOnkoGormun\_IsSurg (N, Н) – Признак «Хирургическая» 0 – нет, 1 – да; * EvnUslugaOnkoGormun\_IsDrug (N, Н) – Признак «Лекарственная» 0 – нет, 1 – да; * EvnUslugaOnkoGormun\_IsOther (N, Н) – Признак «Неизвестно» 0 – нет, 1 – да; * DrugTherapyLineType\_id (N, У) – Линия лекарственной терапии (справочник dbo.DrugTherapyLineType), обязательный, если EvnUslugaOnkoGormun\_IsDrug=2 * DrugTherapyLoopType\_id (N, У) – Цикл лекарственной терапии (справочник dbo.DrugTherapyLoopType), обязательный, если EvnUslugaOnkoGormun\_IsDrug=2 |
| **Ответ** | **Успешный ответ идентификатор записи**   * EvnUslugaOnkoGormun\_id – Ид услуги по ормоноиммунотерапевтическому лечению; |
| **Пример** |  |

* 1. Изменения данных по гормоноиммунотерапевтическому лечению в рамках специфики онкологии

**PUT api/EvnUslugaOnkoGormun**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnUslugaOnkoGormun\_id (N, O) – Ид услуги по ормоноиммунотерапевтическому лечению; * MorbusOnko\_id (N, Н) – Ид заболевания по онкологии. * UslugaComplex\_id (N, Н) – Ид услуги (справочник dbo.UslugaComplex) * EvnUslugaOnkoGormun\_setDT (DT, Н) – Дата начала выполнения услуги; * EvnUslugaOnkoGormun\_disDT (DT, Н) – Дата окончания выполнения услуги; * Lpu\_uid (N, Н) – Место выполнения * OnkoUslugaGormunFocusType\_id (N, Н) – Преимущественная направленность гормоноиммунотерапии (справочник dbo.OnkoUslugaGormunFocusType); * OnkoTreatType\_id (N, Н) – Характер лечения (справочник dbo.OnkoTreatType) * TreatmentConditionsType\_id – Условия проведения (справочник dbo.TreatmentConditionsType); * AggType\_id (N, Н) – Тип осложнения (справочник dbo.AggType); * EvnUslugaOnkoGormun\_IsBeam (N, Н) – Признак «Лучевая» 0 – нет, 1 – да; * EvnUslugaOnkoGormun\_IsSurg (N, Н) – Признак «Хирургическая» 0 – нет, 1 – да; * EvnUslugaOnkoGormun\_IsDrug (N, Н) – Признак «Лекарственная» 0 – нет, 1 – да * EvnUslugaOnkoGormun\_IsOther (N, Н) – Признак «Неизвестно» 0 – нет, 1 – да * DrugTherapyLineType\_id (N, У) – Линия лекарственной терапии (справочник dbo.DrugTherapyLineType), обязательный, если EvnUslugaOnkoGormun\_IsDrug=2 * DrugTherapyLoopType\_id (N, У) – Цикл лекарственной терапии (справочник dbo.DrugTherapyLoopType), обязательный, если EvnUslugaOnkoGormun\_IsDrug=2 |
| **Ответ** | Успешный ответ с ошибкой «0» |
| **Пример** |  |

* 1. Получение данных по гормоноиммунотерапевтическому лечению в рамках специфики онкологии

**GET api/EvnUslugaOnkoGormun**

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusOnko\_id (N, Н) – Ид заболевания по онкологии. |
| **Ответ** | Успешный ответ:   * MorbusOnko\_id (N, Н) – Ид заболевания по онкологии. * EvnUslugaOnkoGormun\_id – Ид услуги по ормоноиммунотерапевтическому лечению; * EvnUsluga\_id – Ид услуги; * Evn\_setDT – Дата начала выполнения услуги; * Evn\_disDT – Дата окончания выполнения услуги; * OnkoUslugaGormunFocusType\_id – Преимущественная направленность гормоноиммунотерапии; * TreatmentConditionsType\_id – Условия проведения; * AggType\_id – Тип осложнения; * EvnUslugaOnkoGormun\_IsBeam – Признак «Лучевая» 0 – нет, 1 – да; * EvnUslugaOnkoGormun\_IsSurg – Признак «Хирургическая» 0 – нет, 1 – да; * EvnUslugaOnkoGormun\_IsDrug – Признак «Лекарственная» 0 – нет, 1 – да; * EvnUslugaOnkoGormun\_IsOther – Признак «Неизвестно» 0 – нет, 1 – да; * OnkoTreatType\_id – Характер лечения. |
| **Пример** |  |

* 1. Создание данных по хирургическому лечению в рамках специфики онкологии

**POST api/EvnUslugaOnkoSurg**

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusOnko\_id (N, Н) – Ид заболевания по онкологии. * UslugaComplex\_id (N, O) – Название операции (справочник dbo.UslugaComplex) * Person\_id (N, Н) – Ид человека; * EvnUslugaOnkoSurg\_setDT (DT, O) – Дата начала выполнения услуги; * EvnUslugaOnkoSurg\_disDT (DT, H) – Дата окончания выполнения услуги; * Lpu\_uid (N, Н) – Место выполнения * OperType\_id (N, Н) – Тип операции (справочник dbo.OperType); * TreatmentConditionsType\_id (N, Н) – Условия проведения (справочник dbo.TreatmentConditionsType). * OnkoSurgTreatType\_id (N, Н) – Характер хирургического лечения (справочник dbo.OnkoSurgTreatType); * OnkoSurgicalType\_id (N, О) – Тип лечения (справочник dbo.OnkoSurgicalType) * AggType\_id (N, Н) – Интраоперационное осложнение (справочник dbo.AggTypev); * AggType\_sid (N, Н) – Послеоперационное осложнение (справочник dbo.AggTypev); |
| **Ответ** | **Успешный ответ идентификатор записи**   * EvnUslugaOnkoSurg\_id – Ид услуги по хирургическому лечению; |

* 1. Изменения данных по хирургическому лечению в рамках специфики онкологии

**PUT api/EvnUslugaOnkoSurg**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnUslugaOnkoSurg\_id (N, O) – Ид услуги по хирургическому лечению; * MorbusOnko\_id (N, Н) – Ид заболевания по онкологии. * UslugaComplex\_id (N, Н) – Название операции (справочник dbo.UslugaComplex) * Person\_id (N, Н) – Ид человека; * EvnUslugaOnkoSurg\_setDT (DT, Н) – Дата начала выполнения услуги; * EvnUslugaOnkoSurg\_disDT (DT, H) – Дата окончания выполнения услуги; * Lpu\_uid (N, Н) – Место выполнения * OperType\_id (N, Н) – Тип операции (справочник dbo.OperType); * TreatmentConditionsType\_id (N, Н) – Условия проведения (справочник dbo.TreatmentConditionsType). * OnkoSurgTreatType\_id (N, Н) – Характер хирургического лечения (справочник dbo.OnkoSurgTreatType); * OnkoSurgicalType\_id (N, Н) – Тип лечения (справочник dbo.OnkoSurgicalType) * AggType\_id (N, Н) – Интраоперационное осложнение (справочник dbo.AggTypev); * AggType\_sid (N, Н) – Послеоперационное осложнение (справочник dbo.AggTypev); |
| **Ответ** | * Успешный ответ с ошибкой «0» |

* 1. Получение данных по хирургическому лечению в рамках специфики онкологии

**GET api/EvnUslugaOnkoSurg**

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusOnko\_id (N, Н) – Ид заболевания по онкологии. |
| **Ответ** | Успешный ответ:   * MorbusOnko\_id (N, Н) – Ид заболевания по онкологии. * EvnUslugaOnkoSurg\_id – Ид услуги по хирургическому лечению; * EvnUsluga\_id – Ид услуги; * Evn\_setDT – Дата начала выполнения услуги; * Evn\_disDT – Дата окончания выполнения услуги; * OperType\_id – Название операции; * OnkoSurgTreatType\_id – Характер хирургического лечения; * AggType\_id – Интраоперационное осложнение; * AggType\_sid – Послеоперационное осложнение; * TreatmentConditionsType\_id – Условия проведения. |

* 1. Создание данных по неспецифическому лечению в рамках специфики онкологии

**POST api/EvnUslugaOnkoNonSpec**

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusOnko\_id (N, Н) – Ид заболевания по онкологии. * UslugaComplex\_id (N, O) – Название услуги (справочник dbo.UslugaComplex) * Lpu\_uid (N, Н) – МО; * EvnUslugaOnkoNonSpec\_setDT (D, O) – Дата выполнения услуги; |
| **Ответ** | **Успешный ответ идентификатор записи**   * EvnUslugaOnkoNonSpec\_id – Ид услуги по неспецифичесому лечению; |

* 1. Изменение данных по неспецифическому лечению в рамках специфики онкологии

**PUT api/EvnUslugaOnkoNonSpec**

|  |  |
| --- | --- |
| **Входящие параметры** | * EvnUslugaOnkoNonSpec\_id (N , O) – Ид услуги по неспецифичесому лечению; * MorbusOnko\_id (N, Н) – Ид заболевания по онкологии; * UslugaComplex\_id (N, У) – Название услуги (справочник dbo.UslugaComplex); * EvnUslugaOnkoNonSpec\_setDT (D, У) – Дата выполнения услуги; * Lpu\_uid (N, Н) – МО |
| **Ответ** | * Успешный ответ с ошибкой «0» |
| **Пример** |  |

* 1. Получение данных по неспецифическому лечению в рамках специфики онкологии

**GET api/Evn UslugaOnkoNonSpec**

|  |  |
| --- | --- |
| **Входящие параметры** | * MorbusOnko\_id (N, Н) – Ид заболевания по онкологии. |
| **Ответ** | Успешный ответ:   * MorbusOnko\_id (N, Н) – Ид заболевания по онкологии; * EvnUslugaOnkoNonSpec\_id – Ид услуги по неспецифичесому лечению; * EvnUsluga\_id – Ид услуги; * Evn\_setDT – Дата выполнения услуги; |

1. Описание методов работы с данными СУР
2. 1. Получение идентификаторов СУР

**Get api/GetPersonalHistoryWP**

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N,O) –идентификатор МО * WorkPlace\_id (N,H) – идентификатор записи «Место работы» |
| **Ответ** | В ответе возвращается соответствующая запись СУР со следующими параметрами:   * PersonalID (N,H) – идентификатор сотрудника (СУР). Возвращается если во входящих параметрах указано WorkPlace\_id (N,H) – идентификатор записи «Место работы» * FPID (N,Н) –идентификатор функционального подразделения сотрудника (СУР). Возвращается если во входящих параметрах указано WorkPlace\_id (N,H) – идентификатор записи «Место работы». * MOID (N,O) –идентификатор МО * SpecialityID (N,O) - Идентификатор специальности |

1. Описание методов работы с прочими объектами
2. 1. Получение показателей крови человека

**GET api/PersonBloodGroup**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, О) – Ид человека |
| **Ответ** | Успешный ответ   * Person\_id (N, О) – Ид человека; * PersonBloodGroop\_id (N, О) – Ид значения показателей; * BloodGroopType\_id (N, Н) – Группа крови; * RhFactorType\_id (N, Н) – Резус-фактор; * PersonBloodGroop\_setDT (D, О) – Дата определния |

* 1. Получение атрибутов МЭС (КЗГ)

**GET api/MesTariff**

|  |  |
| --- | --- |
| **Входящие параметры** | * Mes\_id (N, О) – Ид МЭС (КЗГ) |
| **Ответ** | Успешный ответ   * Mes\_id – Ид МЭС (КЗГ); * MesTariff\_id – Ид тарифа; * MesTariff\_Value – Значение тарифа; * MesPayType\_id – Ид вида тарифа МЭС (КЗГ) |

* 1. Получение данных по амбулаторной карте по пациенту и МО

**GET api/PersonAmbulatCard**

Получение информации по амбулатроной карте по пациенту и МО.

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N, O) – Идентификатор пациента; * Lpu\_id (N, H) – Идентификатор МО. |
| **Ответ** | Успешный ответ список амбулаторных карт с данными:   * Lpu\_id (N, H) – Идентификатор МО прикрепления; * PersonAmbulatCard\_id (N, О) – Идентификатор амбулаторной карты; * PersonAmbulatCard\_Num (N, О) – Номер амбулаторной карты; * PersonAmbulatCard\_begDate (N, О) – Дата открытия карты; * PersonAmbulatCard\_endDate (N, Н) – Дата закрытия карты; * LpuAttachType\_id (N, O) – Идентификатор типа прикрепления к МО. |

* 1. Получение информации по стационарной бирке

**GET api/TimeTableStacById**

|  |  |
| --- | --- |
| **Запрос** | * TimeTableStac\_id (N, O) – идентификатор бирки по стационару |
| **Ответ** | Успешный ответ:   * EvnDirection\_id (N, H) – Идентификатор направления; * Evn\_id (N, H) – Идентификатор случая выписки направления (значение сущности dbo.Evn); * Evn\_pid (N, Н) – Идентификатор случая-родителя (движение в КВС, посещения в ТАП) (значение сущности dbo.Evn); * Person\_id (N, H) – Человек (значение сущности dbo.Person); * LpuSectionBedType\_id (N, О) – Идентификатор типа койки; * TimeTableStac\_setDate (D, О) – Дата начала госпитализации; * TimeTableType\_id (N, H) – Идентификатор типа бирки; * LpuSection\_id (N, О) – Идентификатор отделения МО |

* 1. Получение значениий атрибутов тарифа

**GET api/AttributeValue**

|  |  |
| --- | --- |
| **Входящие параметры** | * TariffClass\_id (Т, Н) – Ид тарифа; * TarifClass\_sysNick (T, Н) – Код тарифа; * TariffClass\_Name (T, Н) – Наименование тарифа; * Attribute\_id (N, Н) – Ид атрибута; * Attribute\_sysNick (T, Н) – Код атрибута; * Attribute\_Name (T, Н) – Наименование атрибута; * Date\_DT (D, Н) |
| **Ответ** | Проверка: один из параметров TariffClass\_Name, TariffClass\_id, TarifClass\_sysNick должен быть заполнен. Иначе ошибка.  Учитывается период действия значения атрибута: значение параметра Date\_DT (если заполнен) должно входить в период между AttributeValue\_begDate и AttributeValue\_endDate.  Успешный ответ – список значений атрибутов с параметрами:   * TariffClass\_id (Т, Н) – Ид тарифа; * TarifClass\_sysNick (T, Н) – Код тарифа; * TariffClass\_Name (T, Н) – Наименование тарифа; * Attribute\_id (N, Н) – Ид атрибута; * Attribute\_sysNick (T, Н) – Код атрибута; * AttributeValue\_begDate (D, Н) – Дата начала действия атрибута; * AttributeValue\_endDate (D, Н) – Дата окончания действия атрибута; * AttributeValue\_ValueFloat (N, Н) – Значение атрибута. |

* 1. Получение МО по идентификатору

**GET api/MOById**

|  |  |
| --- | --- |
| **Входящие параметры** | * Lpu\_id (N, О) – Идентификатор МО |
| **Ответ** | Успешный ответ   * Org\_Name (T, O) – Наименование медицинской организации * Org\_Nick (T, O) – Краткое наименование медицинской организации |

* 1. Получение Таблицы связка регионального и ФЕД профиля коек

**GET api/LpuSectionBedProfileLinkFed**

|  |  |
| --- | --- |
| **Входящие параметры** | * Нет |
| **Ответ** | Успешный ответ   * LpuSectionBedProfileLink\_id – идентификатор записи. * LpuSectionBedProfile\_id – идентификатор профиля коек. * LpuSectionBedProfile\_fedid – идентификатор федерального профиля коек. |

1. Описание методов модуля «Имуннопрофилактика»

Описание методов работы с журналом вакцинации

* 1. Описание создания записи журнала вакцинации

**POST api/vacJournalAccount**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N,O) – уникальный идентификатор человека * vacJournalAccount\_DatePurpose(N,D) дата назначения прививки (при назначении прививки) * vacJournalAccount\_Purpose\_MedPersonal\_id (N,H) ID врача, назначившего прививку * Vaccine\_id (N,О) Идентификатор вакцины (ссылка нас правочник вакцин (vac.S\_Vaccine) * vacJournalAccount\_Seria (N,H) - Серия препарата * vacJournalAccount\_Period (D,H) – срок годности * vacJournalAccount\_Dose (T,H) доза препората (пример заполнения: 0.5мл, 0.05мг * VaccineWay\_id (N,H) – способ введения Ссылка на справочник vac.S\_VaccineWay * VaccinePlace\_id (N,H) место введнеия. Ссылка на справочник места введения вакцины S\_VaccinePlace * vacJournalAccount\_DateVac (D,О) – дата вакцинации * vacJournalAccount\_Lpu\_id (N,H) – МО, где прививка была исполнена * vacJournalAccount\_Vac\_MedPersonal\_id (N,H) Данные о медициском сотруднике, исполнившим прививку |
| **Ответ** | Производятся следующие проверки:   1. Если по переданному значению Person\_id не найден пациент, то ошибка «Пациент не найден в системе». 2. Если привика с аналогичными параметрами найдена в системе, то ошибка «Данные прививки не прошли проверку на дублирование». Создание записи не производится. 3. Если в любом передаваемом параметре передается значение, не соответствующее ожидаемому формату, например:  * ожидается числовое значение, а передано текстовое значение; * ожидается параметр, состоящий из 11 символов, а передан параметр, состоящий из большего или меньшего количества символов;   то, ошибка: «Неверный формат поля <Наименования полей через запятую>», создание записи в БД не происходит.   1. Если в передаваемом параметре (vacJournalAccount\_Purpose\_MedPersonal\_id, Vaccine\_id, VaccineWay\_id, VaccinePlace\_id, vacJournalAccount\_Lpu\_id, vacJournalAccount\_Vac\_MedPersonal\_id) передается значение id, не найденное в системе, ошибка: «Значения для <Наименования полей через запятую> не найдены в справочниках системы». Создание записи в БД не происходит.   Успешный ответ:   1. vacJournalAccount\_id – уникальный идентификатор исполненной приивки |

* 1. Описание изменения записи журнала вакцинации

**PUT api/vacJournalAccount**

|  |  |
| --- | --- |
| **Входящие параметры** | * vacJournalAccount\_id (N,O) – уникальный идентификатор исполненной приивки * Person\_id (N,H) – уникальный идентификатор человека * vacJournalAccount\_DatePurpose(D,H) дата назначения прививки (при назначении прививки) * vacJournalAccount\_Purpose\_MedPersonal\_id (N,H) ID врача, назначившего прививку * Vaccine\_id (N,H) Идентификатор вакцины (ссылка нас правочник вакцин (vac.S\_Vaccine) * vacJournalAccount\_Seria (N,H) - Серия препарата * vacJournalAccount\_Period (D,H) – срок годности * vacJournalAccount\_Dose (T,H) доза препората (пример заполнения: 0.5мл, 0.05мг * VaccineWay\_id (N,H) – способ введения Ссылка на справочник vac.S\_VaccineWay * VaccinePlace\_id (N,H) место введнеия. Ссылка на справочник места введения вакцины S\_VaccinePlace * vacJournalAccount\_DateVac (D,H) – дата вакцинации * vacJournalAccount\_Lpu\_id (N,H) – МО, где прививка была исполнена * vacJournalAccount\_Vac\_MedPersonal\_id (N,H) Данные о медициском сотруднике, исполнившим прививку |
| **Ответ** | Производятся следующие проверки:   1. Если по переданному значению vacJournalAccount\_id не найдена прививка, то ошибка «Прививка не найдена в системе». 2. Если по переданному значению Person\_id не найдена vacJournalAccount\_id, то ошибка «Данная прививка для указанного пациента не найдена в системе». 3. Если привика с аналогичными параметрами (с учетом изменений по передаваемым параметрам) найдена в системе, то ошибка «Данные прививки не прошли проверку на дублирование». Создание записи не производится. 4. Если в любом передаваемом параметре передается значение, не соответствующее ожидаемому формату, например:  * ожидается числовое значение, а передано текстовое значение; * ожидается параметр, состоящий из 11 символов, а передан параметр, состоящий из большего или меньшего количества символов;   то, ошибка: «Неверный формат поля <Наименования полей через запятую>», обновление записи в БД не происходит.   1. Если в передаваемом параметре (vacJournalAccount\_Purpose\_MedPersonal\_id, Vaccine\_id, VaccineWay\_id, VaccinePlace\_id, vacJournalAccount\_Lpu\_id, vacJournalAccount\_Vac\_MedPersonal\_id) передается значение id, не найденное в системе, ошибка: «Значения для <Наименования полей через запятую> не найдены в справочниках системы». Обновление записи в БД не происходит.   При успешной обработке запроса возвращается ответ с кодом ошибки «0» |

* 1. Описание получения записи журнала вакцинации

**GET api/vacJournalAccount**

|  |  |
| --- | --- |
| **Входящие параметры** | * vacJournalAccount\_id (N,H) – уникальный идентификатор исполненной приивки * Person\_id (N,H) – уникальный идентификатор человека |
| **Ответ** | Если в результате обработки запроса не было найдено ни одной записи , то ответ пустой  Если в параметрах запроса не указан хотя бы один параметр, то ошибка  Если в результате обработки запроса была найдена хотя бы одна запись, то в ответ список со следующими параметрами:   * vacJournalAccount\_id (N,O) – уникальный идентификатор исполненной приивки * Person\_id (N,O) – уникальный идентификатор человека * vacJournalAccount\_DatePurpose(N,D) дата назначения прививки (при назначении прививки) * vacJournalAccount\_Purpose\_MedPersonal\_id (N,H) ID врача, назначившего прививку * Vaccine\_id (N,О) Идентификатор вакцины (ссылка нас правочник вакцин (vac.S\_Vaccine) * vacJournalAccount\_Seria (N,H) - Серия препарата * vacJournalAccount\_Period (D,H) – срок годности * vacJournalAccount\_Dose (T,H) доза препората (пример заполнения: 0.5мл, 0.05мг * VaccineWay\_id (N,H) – способ введения Ссылка на справочник vac.S\_VaccineWay * VaccinePlace\_id (N,H) место введнеия. Ссылка на справочник места введения вакцины S\_VaccinePlace * vacJournalAccount\_DateVac (D,О) – дата вакцинации * vacJournalAccount\_Lpu\_id (N,H) – МО, где прививка была исполнена * vacJournalAccount\_Vac\_MedPersonal\_id (N,H) Данные о медициском сотруднике, исполнившим прививку |

Описание методов работы с туберкулиновой пробой

* 1. Описание создания записи о туберкулиновой пробе

**POST api/vac\_JournalMantu**

|  |  |
| --- | --- |
| **Входящие параметры** | * Person\_id (N,O) – уникальный идентификатор человека * JournalMantu\_Dose (T,H) – доза препарата * VaccineWay\_id (T, О) способ введения Ссылка на справочник vac.S\_VaccineWay.. * VaccinePlace\_id(N, О) место введнеия. Ссылка на справочник места введения вакцины S\_VaccinePlace. * MantuReactionType\_id (N,H) – тип реакции манту. Ссылка на справочинк vac.S\_MantuReactionType * JournalMantu\_DateVac (N,O) – дата пробы * JournalMantu\_Lpu\_id (N,H) – идентификатор МО, в которой взяли пробу * JournalMantu\_vacMedPersonal\_id (N, О) – идентификатор медицинского сотрудника, исполнившего прививку. * TubDiagnosisType\_id(N,О) – тип туберкулиновой пробы. Справочник vac.S\_TubDiagnosisType * DiaskinTestReactionType\_id (N,H) – реакция диаскин тест . Ссылка на справочинк vac.S\_DiaskinTestReactionType * JournalMantu\_Seria (S, O) – Серия вакцины * JournalMantu\_Period (D, O)– Срок годности вакцины |
| **Ответ** | Производятся следующие проверки:   1. Если по переданному значению Person\_id не найден пациент, то ошибка «Пациент не найден в системе». 2. Если привика с аналогичными параметрами найдена в системе, то ошибка «Данные привки не прошли проверку на дублирование». Создание записи не производится. 3. Если в любом передаваемом параметре передается значение, не соответствующее ожидаемому формату, например:  * ожидается числовое значение, а передано текстовое значение; * ожидается параметр, состоящий из 11 символов, а передан параметр, состоящий из большего или меньшего количества символов;   то, ошибка: «Неверный формат поля <Наименования полей через запятую>», создание записи в БД не происходит.   1. Если в передаваемом параметре (VaccineWay\_id, VaccinePlace\_id, MantuReactionType\_id, JournalMantu\_Lpu\_id, JournalMantu\_vacMedPersonal\_id, TubDiagnosisType\_id, DiaskinTestReactionType\_id) передается значение id, не найденное в системе, ошибка: «Значения для <Наименования полей через запятую> не найдены в справочниках системы». Создание записи в БД не происходит.   Успешный ответ:   1. JournalMantu\_id (N,O) – уникальный идентификатор зписи |

* 1. Описание изменения записи о туберкулиновой пробе

**PUT api/vac\_JournalMantu**

|  |  |
| --- | --- |
| **Входящие параметры** | * JournalMantu\_id (N,O) – уникальный идентификатор зписи * Person\_id (N,H) – уникальный идентификатор человека * JournalMantu\_Dose (T,H) – доза препарата * VaccineWay\_id (T,H) способ введения Ссылка на справочник vac.S\_VaccineWay * VaccinePlace\_id(N,H) место введнеия. Ссылка на справочник места введения вакцины S\_VaccinePlace * MantuReactionType\_id (N,H) – тип реакции манту. Ссылка на справочинк vac.S\_MantuReactionType * JournalMantu\_DateVac (N,H) – дата пробы * JournalMantu\_Lpu\_id (N,H) – идентификатор МО, в которой взяли пробу * JournalMantu\_vacMedPersonal\_id (N,H) – идентификатор медицинского сотрудника, исполнившего пививку * TubDiagnosisType\_id(N,H) - тип туберкулиновой пробы. Справочник vac.S\_TubDiagnosisType * DiaskinTestReactionType\_id (N,H) – реакция диаскин тест . Ссылка на справочинк vac.S\_DiaskinTestReactionType |
| **Ответ** | Производятся следующие проверки:   1. Если по переданному значению JournalMantu\_id не найдена прививка, то ошибка «Прививка не найдена в системе». 2. Если по переданному значению Person\_id не найдена JournalMantu\_id, то ошибка «Данная прививка для указанного пациента не найдена в системе». 3. Если привика с аналогичными параметрами найдена в системе, то ошибка «Данные привки не прошли проверку на дублирование». Создание записи не производится. 4. Если в любом передаваемом параметре передается значение, не соответствующее ожидаемому формату, например:  * ожидается числовое значение, а передано текстовое значение; * ожидается параметр, состоящий из 11 символов, а передан параметр, состоящий из большего или меньшего количества символов;   то, ошибка: «Неверный формат поля <Наименования полей через запятую>», создание записи в БД не происходит.   1. Если в передаваемом параметре (VaccineWay\_id, VaccinePlace\_id, MantuReactionType\_id, JournalMantu\_Lpu\_id, JournalMantu\_vacMedPersonal\_id, TubDiagnosisType\_id, DiaskinTestReactionType\_id) передается значение id, не найденное в системе, ошибка: «Значения для <Наименования полей через запятую> не найдены в справочниках системы». Создание записи в БД не происходит.   При успешной обработке запроса возвращается ответ с кодом ошибки «0» |

* 1. Описание получения записей о туберкулиновых пробах

**GET api/vac\_JournalMantu**

|  |  |
| --- | --- |
| **Входящие параметры** | * JournalMantu\_id (N,Н) – уникальный идентификатор зписи * Person\_id (N,H) – уникальный идентификатор человека |
| **Ответ** | Если в результате обработки запроса не было найдено ни одной записи , то ответ пустой  Если в параметрах запроса не указан хотя бы один параметр, то ошибка  Если в результате обработки запроса была найдена хотя бы одна запись, то в ответ список со следующими параметрами:   * JournalMantu\_id (N,O) – уникальный идентификатор зписи * Person\_id (N,H) – уникальный идентификатор человека * JournalMantu\_Dose (T,H) – доза препарата * VaccineWay\_id (T,H) способ введения Ссылка на справочник vac.S\_VaccineWay * VaccinePlace\_id(N,H) место введнеия. Ссылка на справочник места введения вакцины S\_VaccinePlace * MantuReactionType\_id (N,H) – тип реакции манту. Ссылка на справочинк vac.S\_MantuReactionType * JournalMantu\_DateVac (N,H) – дата пробы * JournalMantu\_Lpu\_id (N,H) – идентификатор МО, в которой взяли пробу * JournalMantu\_vacMedPersonal\_id (N,H) – идентификатор медицинского сотрудника, исполнившего пививку * TubDiagnosisType\_id(N,H) - тип туберкулиновой пробы. Справочник vac.S\_TubDiagnosisType * DiaskinTestReactionType\_id (N,H) – реакция диаскин тест . Ссылка на справочинк vac.S\_DiaskinTestReactionType |

1. Описание методов работы с ресурсами
   1. Добавление нового ресурса

**POST api/Resource**

|  |  |
| --- | --- |
| **Входящие параметры** | * Resource\_Name (S, О) – наименование ресурса * ResourceType\_id (N, О) – идентификатор типа ресурса (1 –Общий, 2-операционный стол, 3- Аппарат) * Resource\_begDT (D, О) – дата начала * Resource\_endDT (D, N) – дата окончания * MedService\_id(N, О) – идентификатор службы ресурса * Массив карточек мед. изделий MedProductCardResourceData{ * MedProductCard\_id(N, У), – идентификаторы карточек медицинских изделий, связанных с ресурсом. Обязательный параметр, если ResourceType\_id = 3 * begDT(D, О) – дата начала * endDT(D, N) – дата окончания} |
| **Ответ** | Если указанResourceType\_id = 3 и передан пустой массив MedProductCardResourceData, то выводится ошибка.  Если указанResourceType\_id = 3 и передан НЕ пустой массив MedProductCard\_id, то выполняется проверка на пересечение периодов связи карточек мед. изделий с ресурсом (в один и тот-же момент времени ресурс не может быть связан более чем с одним мед.изделием). Если обнаружено пересечение, то возвращается ошибка, иначе создаются связи создаваемого ресурса и мед. изделий – записи в таблице passport.MedProductCardResource  **Успешный ответ**  Возвращает идентификатор ресурса   * Resource \_id (N, О) – идентификатор ресурса |

* 1. Редактирование ресурса

**PUT api/Resource**

|  |  |
| --- | --- |
| **Входящие параметры** | * Resource\_id (N,O) – идентификатор ресурса * Resource\_Name (S, О) – наименование ресурса * ResourceType\_id (N, О) – идентификатор типа ресурса (1 –Общий, 2-операционный стол, 3- Аппарат) * Resource\_begDT (D, О) – дата начала * Resource\_endDT (D, N) – дата окончания * Массив карточек мед. изделий (обязательный параметр, если ResourceType\_id = 3), MedProductCardResourceData{ * MedProductCard\_id(N, О) – идентификаторы карточек медицинских изделий, связанных с ресурсом. * begDT(D, О) – дата начала * endDT(D, О) – дата окончания} |
| **Ответ** | * У ресурса с идентификатором Resource\_id параметры Resource\_Name, Resource\_begDT, Resource\_endDT меняются на соответствующие входящие параметры метода * Выполняется проверка на смену типа ресурса:  1. Новый ResourceType\_id = 3, старый ResourceType\_id = 3  * **Если** передан Массив карточек мед. изделий MedProductCardResourceData, **то** старые связи ресурса с мед изделиями удаляются и создаются новые связи ресурса с указанными изделиями **иначе** ошибка  1. Новый ResourceType\_id <> 3, старый ResourceType\_id = 3  * Удаляются связи ресурса с мед изделиями * Тип ресурса меняется на новый  1. Новый ResourceType\_id = 3, старый ResourceType\_id <> 3  * **Если** передан Массив карточек мед. изделий, **то** создаются новые связи ресурса с указанными изделиями **иначе** ошибка   **Успешный ответ**  Возвращается идентификатор ресурса  Resource\_id (N,O) – идентификатор ресурса |

* 1. Добавление расписания на ресурс

**POST api/TimeTableResource/TimeTableResource**

|  |  |
| --- | --- |
| **Входящие параметры** | * Resource\_id (N, O) – идентификатор ресурса   Массив данных для создания бирок: TimeTableResourceCreate{   * TimeTablResource\_begTime (DT, O) – Дата и время начала приема * TimeTableResource\_Time (N,O) – Длительность приема * TimeTableType\_id (N, O) – идентификатор типа бирки. dbo.TimeTableType * TimeTableResource\_IsDop (N, О) – признак дополнительной бирки. Возможные значения: 0 и 1, - где 0 – нет, 1 – да.   } |
| **Ответ** | Успешный ответ – список бирок   * TimeTableResource\_id (N, O) - Идентификатор бирки |

* 1. Изменение расписания на ресурс

**PUT api/TimeTableResource/TimeTableResource**

|  |  |
| --- | --- |
| **Входящие параметры** | * Resource\_id (N, O) – идентификатор ресурса   Массив данных для создания бирок: TimeTableResourceEdit{   * TimeTableResource\_id (N, O) – идентификатор бирки * [TimeTableType](http://fixuploader.swn.local/dbspr/index/627845649)\_id (N, H) –тип бирки, на которую нужно изменить. Значение из справочника dbo.TimeTableType * TimeTableResourceDelStatus (N, H) – признак удаления бирки * TimeTableResource\_IsDop (N,H) – признак дополнительной бирки   } |
| **Ответ** | Успешный ответ – пустой ответ с кодом ошибки «0» |

1. Импорт данных из СЭМД «Направление на госпитализацию, восстановительное лечение, обследование, консультацию» (ИЭМК)
   1. Схема импорта направлений

Загрузка в РИШ из сторонней системы данных СЭМД «Направление на госпитализацию, восстановительное лечение, обследование, консультацию» (ИЭМК) осуществляется с помощью rest-методов.

Порядок обработки входящего документа:

1. Обработывается информации о человеке. Вызываются rest-методы добавления человека при передаче данных из ИЭМК.
2. Создается или редактируется запись о человеке с параметом person\_id.
3. Обрабатывается информация о направлении. Вызываются rest-методы для работы с направлениями: GET api/EvnDirection (см. раздел 4.9.2). Используются входные параметры: Person\_id, EvnDirection\_Num, EvnDirection\_setDate.
4. Если направление с такими параметрами не найдено, то вызывается метод создания направления POST api/EvnDirection (см. раздел 4.9.3). Если найдено направление с такими параметрами, то метод возвращает ошибку и направление не создается. Передаваемые параметры и правила заполнения приведены в таблице Таблица 2.

Таблица 3 – Заполнение параметров метода по созданию направления

| **Параметр rest метода** | **Правила заполнения** | **Комментарий** |
| --- | --- | --- |
| Evn\_pid (N, Н) | Не заполняется |  |
| Person\_id (N, О) | Параметр, полученный после успешной обработки информации о человеке |  |
| EvnDirection\_Num (N, О) | Корневой элемент <ClinicalDocument>  id root extension  Пример: <id root="1.2.643.5.1.13.13.12.2.77.7823.100.1.1.51" extension="7133"/> | Уникальный идентификатор документа (данные из extension) |
| EvnDirection\_setDate (D, О) | Корневой элемент <ClinicalDocument>  effectiveTime value | Заполняется параметром, указанным в «Дата создания документа» |
| PayType\_id (N, Н) | Не заполняется |  |
| DirType\_id (N, О) | Секция "Цель направления и медицинская организация, куда направлен" SCOPORG  Элемент entry/act  атрибут @code | Для определения используется таблица НСИ «Тип направления» (OID 1.2.643.5.1.13.13.11.1009)  (fed.DirType) по полученному коду определяется DirType\_id (ссылка на dbo.DirType) |
| Diag\_id (N, О) | Секция "Диагнозы" DGN  Элемент entry/observation  @code | Сохраняется ссылка на Diag\_id по соответствующему коду МКБ-10 |
| EvnDirection\_Descr (S, Н) | Секция "Описание и обоснование направления" ELU |  |
| Lpu\_sid (N, О) – Направившее МО | Элемент ClinicalDocument/recordTarget/patientRole/providerOrganization/id  Атрибут @root | Содержит OID направляющей организации из справочника НСИ "Регистр медицинских организаций Российской Федерации. Версия 2" (идентификатор справочника 1.2.643.5.1.13.2.1.1.178)  Вызов метода Get api/Refbook/Lpu с параметрами:  • LPU\_OID = root  В ответе метода вернется параметр Lpu\_id - это значение передавать в Lpu\_sid |
| LpuSection\_id (N, O) | Не заполняется |  |
| MedPersonal\_id (N, O) | Не заполняется |  |
| MedStaffFact\_id (N, O) | Не заполняется |  |
| MedPersonal\_zid (N, Н) | Не заполняется |  |
| Lpu\_did (N, О) | Секция "Цель направления и медицинская организация, куда направлен" SCOPORG  Элемент  entry/act/performer/assignedEntity/representedOrganization/id  Атрибут @root | Вызов метода Get api/Refbook/Lpu с параметрами:  • LPU\_OID = root  В ответе метода вернется параметр Lpu\_id - это значение передавать в Lpu\_did |
| LpuUnit\_did (N, O) | Не заполняется |  |
| LpuSectionProfile\_id (N, O) | Не заполняется |  |
| MedPersonal\_did (N, Н) | Не заполняется |  |
| TimeTableGraf\_id (N, H) | Не заполняется |  |
| TimeTableStac\_id (N, H) | Не заполняется |  |
| TimeTableMedService\_id (N, H) | Не заполняется |  |
| TimeTableResource\_id (N, H) | Не заполняется |  |
| PrescriptionType\_id (N, У) | Не заполняется |  |
| EvnPrescr\_IsCito (N, У) | Не заполняется |  |
| UslugaComplex\_id (N, У) | Секция "Цель направления и медицинская организация, куда направлен" SCOPORG  Элемент entry/act/entryRelationship/act/code  элемент translation  @codeSystem должен быть “1.2.643.5.1.13.13.11.1070” ИЛИ «1.2.643.5.1.13.2.1.1.473»  Атрибут @code | Вызов метода GET api/Refbook с параметрами:  • Refbook\_Code = ‘1.2.643.5.1.13.2.1.1.473’  • Code = code.  В ответе метода вернется параметр Id - это значение передавать в UslugaComplex\_id. |
| MedService\_id (N, H) | Не заполняется |  |
| Resource\_id (N, H) | Не заполняется |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц) в докум. | № документа | Входящий № сопроводи- тельного документа | Подпись | Дата |
| Изм | Изменен- ных | Заменен- ных | новых | Аннули- рованных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |