**ГОСУДАРСТВЕННая ИНФОРМАЦИОННая**

**СИСТЕМа санкт-петербцрга «Региональный фрагмент кдиной государственной информационной ситсемы В СФЕРЕ ЗДРАВООХРАНЕНИЯ»**

**ОПИСАНИЕ ИНТЕГРАЦИОННЫХ ПРОФИЛЕЙ**

**ПОДСИТСЕМЫ «ТЕЛЕМЕДИЦИНА»**

На 43 листах

# **История версий**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Версия** | **Дата** | **Автор** | **Примечание** |
| 1 | 1.1 draft | 20.05.2020 | Турикин А. | Драфт описания интеграционных профилей. В следующей версии документа будет добавлено описание правил получения ссылок на Модуль организации конференцсвязи и Модуль обмена сообщениями (чат) для вывода их пользователю, передачи консультативного заключения по результатам телемедицинской консультации. |

Содержание

[**История версий** 2](#_Toc41316147)

[**Определения, обозначения и сокращения** 4](#_Toc41316148)

[**Общая информация** 5](#_Toc41316149)

[**Описание решения** 7](#_Toc41316150)

[**Внутренняя логика сервиса** 1](#_Toc41316151)

[**Базовый сценарий работы с сервисом** 3](#_Toc41316152)

[**Интеграция с иными сервисами ГИС РЕГИЗ в рамках работы с телемедицинскими заявками** 5](#_Toc41316153)

[**Описание методов обмена сведениями** 6](#_Toc41316154)

[**1.** **Получение списка переходов для создания заявки доступных пользователю** 6](#_Toc41316155)

[**2.** **Создание заявки по маршруту** 9](#_Toc41316156)

[**3.** **Получение списка доступных для действия заявок** 12](#_Toc41316157)

[**4.** **Получение списка доступных для просмотра заявок** 14](#_Toc41316158)

[**5.** **Получение заявки по идентификатору** 17](#_Toc41316159)

[**6.** **Получение объекта контекста процесса** 20](#_Toc41316160)

[**7.** **Получение идентификаторов следующих возможных состояний для процесса** 21](#_Toc41316161)

[**8.** **Получение описания правил перехода по его идентификатору** 23](#_Toc41316162)

[**9.** **Получение схемы данных передаваемых при осуществлении перехода** 26](#_Toc41316163)

[**10.** **Осуществление перехода в указанное состояние в рамках указанного процесса** 31](#_Toc41316164)

[**Коды возвращаемых ошибок** 35](#_Toc41316165)

# **Определения, обозначения и сокращения**

Таблица 1. Определения, обозначения и сокращения

|  |  |
| --- | --- |
| **Термин** | **Содержание** |
| ВКС | Видеоконференцсвязь |
| ГИС РЕГИЗ | Государственная информационная система Санкт-Петербурга «Региональный фрагмент единой государственной информационной системой в сфере здравоохранения» |
| Индекс пациента | Подсистема ГИС РЕГИЗ «Индекс пациента» |
| ИЭМК | Подсистема ГИС РЕГИЗ «Интегрированная электронная медицинская карта петербуржца» |
| МО | Медицинская организация |
| НСИ | Подсистема ГИС РЕГИЗ «Нормативно-справочная информация» |
| ОДИИ | Подсистема ГИС РЕГИЗ «Обмен данными инструментальных исследований» |
| ОДЛИ | Подсистема ГИС РЕГИЗ «Обмен данными лабораторных исследований» |
| ТМК | Телемедицинская консультация – консультация, проводимая в целях оказания медицинской помощи с использованием телекоммуникационных технологий |
| ТМ.РЕГИЗ | Подсистема «Телемедицина» государственной информационной системы «Региональный фрагмент единой государственной информационной системы в сфере здравоохранения» |
| УО | Подсистема ГИС РЕГИЗ «Управление очередями на оказание медицинской помощи» |

# **Общая информация**

Настоящий документ описывает регламент работы с подсистемой «Телемедицина» государственной информационной системы «Региональный фрагмент единой государственной информационной системы в сфере здравоохранения» (ТМ.РЕГИЗ).

Описание предназначено для организаций-разработчиков, осуществляющих сопровождение эксплуатируемых информационных систем и разработку новых систем для медицинских организаций Санкт-Петербурга.

ТМ.РЕГИЗ предназначена для обеспечения получения медицинской помощи в дистанционной форме.

Указанный сервис обеспечивает механизмы взаимодействия и обмена сведениями о заявках на телемедицинские консультации (далее – ТМК), маршрутах их прохождения, результатах обработки заявок и оказания ТМК.

Для организации дистанционного взаимодействия участников телемедицинских консультаций ТМ.РЕГИЗ включает в себя Компонент организации сеансов видеконференцсвязи (ВКС) и Модуль обмена сообщениями (чат), позволяющие в автоматическом режиме создавать чаты и «комнаты» для проведения сеансов ВКС при формировании заявки на  ТМК.

Компонент организации сеансов конференцсвязи включает следующие модули:

* модуль организации конференцсвязи;
* модуль записи и трансляции конференций;
* модуль совместной интерактивной работы на виртуальном холсте;
* модуль публикации удаленного рабочего стола.

Модуль организации конференцсвязи обеспечивает:

* выполнение видеозвонков между пользователями;
* доступ к чатам и инструментам совместной работы из интерфейса конференций;
* смену разрешения видео;
* тестирование оборудования, доступности и характеристик каналов связи;
* управление участниками;
* монитор качества связи у участников.

Модуль записи и трансляции обеспечивает:

* трансляцию конференции в различных разрешениях;
* доступ к архиву записей;
* просмотр и выгрузку записей.

Модуль совместной интерактивной работы на виртуальном холсте обеспечивает:

* наличие общей интерактивной виртуальной доски;
* нанесение графических примитивов и текста;
* загрузку изображений и документов;
* отмену действий.

Модуль обмена сообщениями обеспечивает:

* организацию приватных чатов;
* организацию групповых чатов;
* выдачу уведомления о прочтении;
* поиск по сообщению;
* форматирование сообщений;
* цитирование сообщений;
* передачу в сообщении файлов и документов;
* передачу видеосообщений;
* уведомления о событиях в чатах.

# **Описание решения**

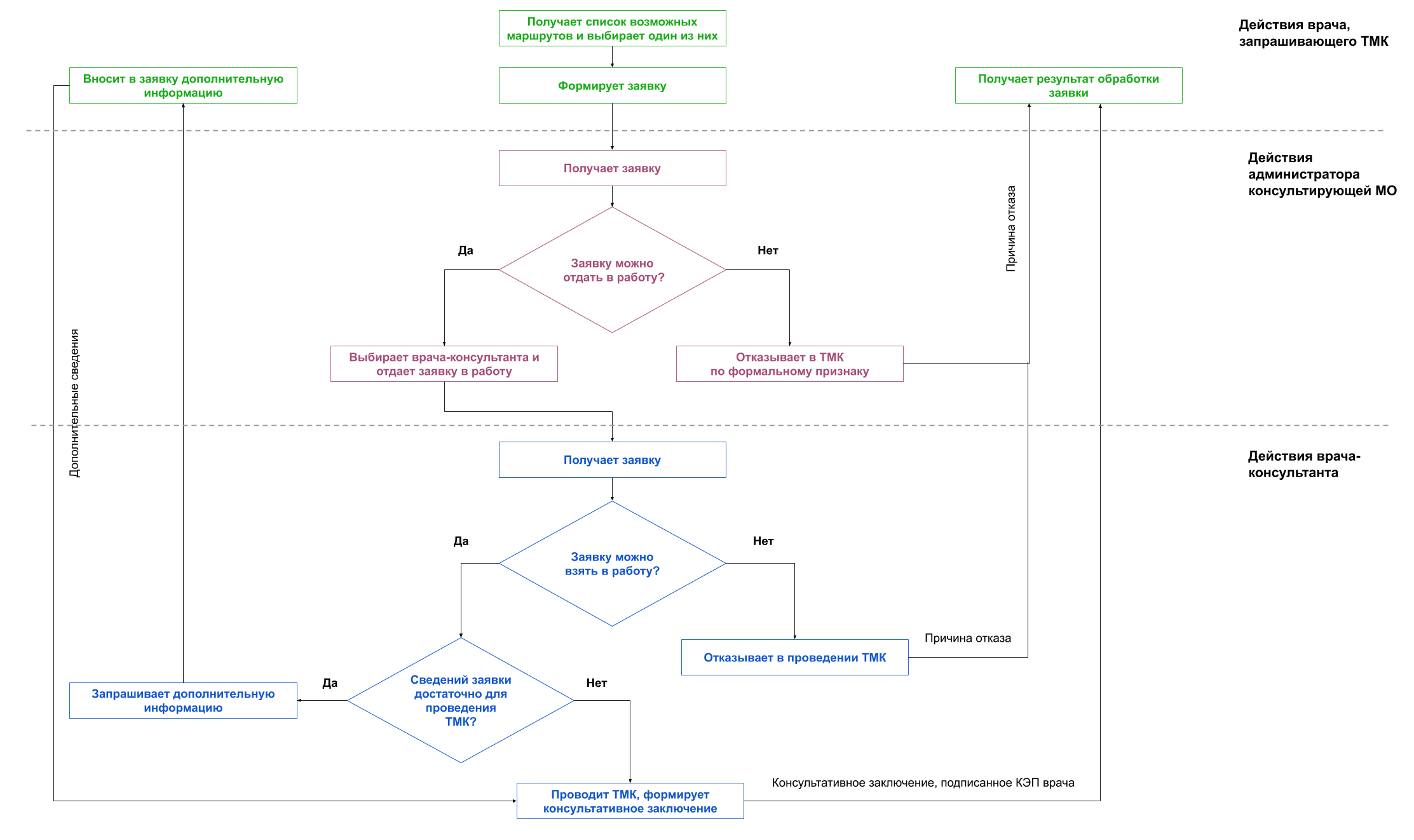
ТМ.РЕГИЗ предназначена для обмена информацией по телемедицинским заявкам. Для разных типов заявок МО может выступать как в роли направляющей МО, так и в роли целевой (консультирующей) МО.

Процесс телемедицинских консультаций, реализуемый в ТМ.РЕГИЗ, осуществляется посредством следующих методов обмена информацией:

1. Получение перечня маршрутов прохождения заявок на ТМК и описания операций, в рамках которых пользователь имеет право создавать заявки.
2. Создание заявки на ТМК по выбранному маршруту.
3. Получение пользователем списка заявок, на ТМК которые имеют операции доступные данному пользователю.
4. Получение пользователем списка заявок на ТМК в статусах, которые доступны данному пользователю для просмотра и действия.
5. Получение сведений заявки на ТМК по её идентификатору.
6. Получение набора данных (объекта контекста) собранного при работе с заявкой на ТМК.
7. Получение перечня переходов (операций), которые доступны пользователю в текущем состоянии заявки на ТМК.
8. Получение условий перехода (выполнения операций) к новому состоянию (статусу) заявки на ТМК.
9. Получение схемы данных (объекта контекста), необходимого для осуществления перехода к новому состоянию (статусу) заявки.
10. Запрос перехода заявки в новое состояние согласно указанному правилу перехода.

ТМ.РЕГИЗ поддерживает возможность реализации различных сценариев работы с заявками на ТМК. Сценарии зависят от набора операций, доступных для заявки на ТМК. Под операциями понимаются различные действия над заявкой, такие как распределение заявки врачу-консультанту, возврат заявки запрашивающему врачу на дополнение, отказ в консультации по заявке, дополнение заявки сведениями, формирование консультативного заключения по заявке и т.д.

Примеры возможного сценария проведения ТМК приведен на рисунке 1.

Рис. 1. Общий сценарий проведения ТМК.

# **Внутренняя логика сервиса**

**Описание данных.**

Структуры данных, которыми системы обмениваются в ходе работы с заявками, описываются с помощью json-схем. Идентификаторы схем содержатся в описании операций (переходов между статусами) над заявками.

Предметная область – сущность, объединяющая под собой описания данных и маршруты бизнес-процессов.

Общий набор данных - json-схема, описывающая все возможные поля и структуры данных, используемых в документообороте. Схемы, описанные ниже, должны быть сконструированы на основе общего набора данных. Это обеспечивает автоматизацию сбора, поиска и агрегации данных. Хранится одна схема на предметную область.

Метаданные маршрутов бизнес-процессов - json-схема описывающая краткий набор ключевых характеристик маршрута бизнес процесса в рамках предметной области. Позволяет выводить контекстную информацию о маршрутах в списке, строить по маршрутам форму фильтрации. Заявки также можно фильтровать по метаданным маршрутов. Хранится одна схема на предметную область.

Метаданные заявок - json-схема, описывающая краткий набор ключевых характеристик заявки. Позволяет выводить контекстную информацию о заявках в списке, строить по заявкам форму фильтрации. Хранится одна схема на предметную область.

Данные, используемые при операциях с заявками - json-схема, описывающая набор и структуру данных необходимых для передачи со стороны пользователя при совершении операции с заявкой. Можно создать в рамках предметной области столько, сколько требуется.

Роли - json-схема, описывающая набор и структуру данных необходимых для передачи информации о пользователе при проверке доступа к функциям системы. Можно создать в рамках предметной области столько, сколько требуется. Пользователь может на входе передавать несколько наборов присвоенных ему ролей. Например, пользователь может быть одновременно врачом в одной организации и административным руководителем в другой. Соответственно заявки и маршруты ему будут подбираться с учетом обеих его ролей.

**Маршрут и его описание.**

Маршрут - описание порядка выполнения операций в ходе исполнения работ по заявкам.

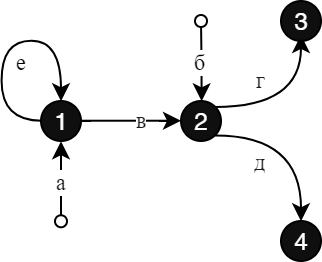


Рис. 2. Схема состояний и операций над заявкой.

Маршрут описывается с помощью названия, описания и структуры метаданных. Суть маршрута заключается в наборе состояний и операций с заявками.

Состояния или статусы (1, 2, 3 на рисунке) - это контрольные точки в бизнес-процессе. Они создаются в рамках маршрута с указанием имени и правил доступа. Правило доступа описывает, какому пользователю может быть доступна заявка для просмотра в данном статусе.

Переходы (а, б, в, г, д, е на рисунке) - это операции над заявкой по данному маршруту.

Если стартовое состояние перехода не указано (а, б на рисунке), то он используется для создания заявки в том статусе, который указан как конечный. Операций создания заявки по маршруту может быть сколько угодно с любой логикой. Например, можно настроить создание заявки со статусом доступным исполнителю. Можно создать заявку в статусе “черновик”, доступным заявителю.

Если стартовое состояние указано тоже, что и конечное (е на рисунке), то операция не изменит статус заявки. Решает задачу редактирования данных заявки без необходимости изменения статуса.

При указании конечного состояния отличного от стартового - заявка меняет состояние (в, г, д на рисунке).

**Переходы и ролевой доступ к просмотру.**

Содержание правила перехода в рамках маршрута:

1. Название операции или перехода.
2. Начальное состояние. Не указывается, если операция предназначена для создания заявки.
3. Итоговое состояние. Указывается обязательно.
4. Набор данных, необходимых для осуществления операции.
5. Проверки. Осуществляют проверки доступа к выполнению операции и другую бизнес-логику, необходимую для поддержки бизнес-процесса. Текстовое описание каждой проверки содержится в описаниях операций.
6. Отклики. Уведомление внешнего сервиса о факте перехода конкретной заявки в определенный статус. Текстовое описание каждого отклика содержится в описаниях операций.

# **Базовый сценарий работы с сервисом**

Формирование маршрутов для заявок происходит на уровне администрирования системы. После того как правила работы с заявками определены в системе, участники взаимодействия должны будут реализовывать свою работу в следующем порядке:

1. Направляющая МО запрашивает из системы список маршрутов и операций, доступных для создания заявки.
2. По полученным данным направляющая МО запрашивает схемы передачи данных, необходимых для осуществления операции создания заявки: формат структуры данных передаваемых в заявку, используемых при описании характеристик пользователя.
3. Направляющая МО создаёт заявку.
4. Целевая (консультирующая) МО запрашивает списки заявок, доступных для выполнения доступных операций согласно их текущему состоянию.
5. По полученным данным целевая МО запрашивает схемы передачи данных необходимых для осуществления операции изменения заявки.
6. Для принятия решения о последующих действиях целевая МО запрашивает содержание заявки.
7. Целевая МО осуществляет операцию изменения состояния заявки. Если бизнес-процесс обработки заявки того требует, то в свою очередь направляющая МО выполняет соответствующие операции обработки заявки. Это происходит пока заявка не придет к конечному состоянию.
8. Направляющая МО запрашивает список заявок, доступных для просмотра.
9. Для получения результата по закрытой заявке, направляющая МО запрашивает содержание заявки. Получает консультативное заключение.

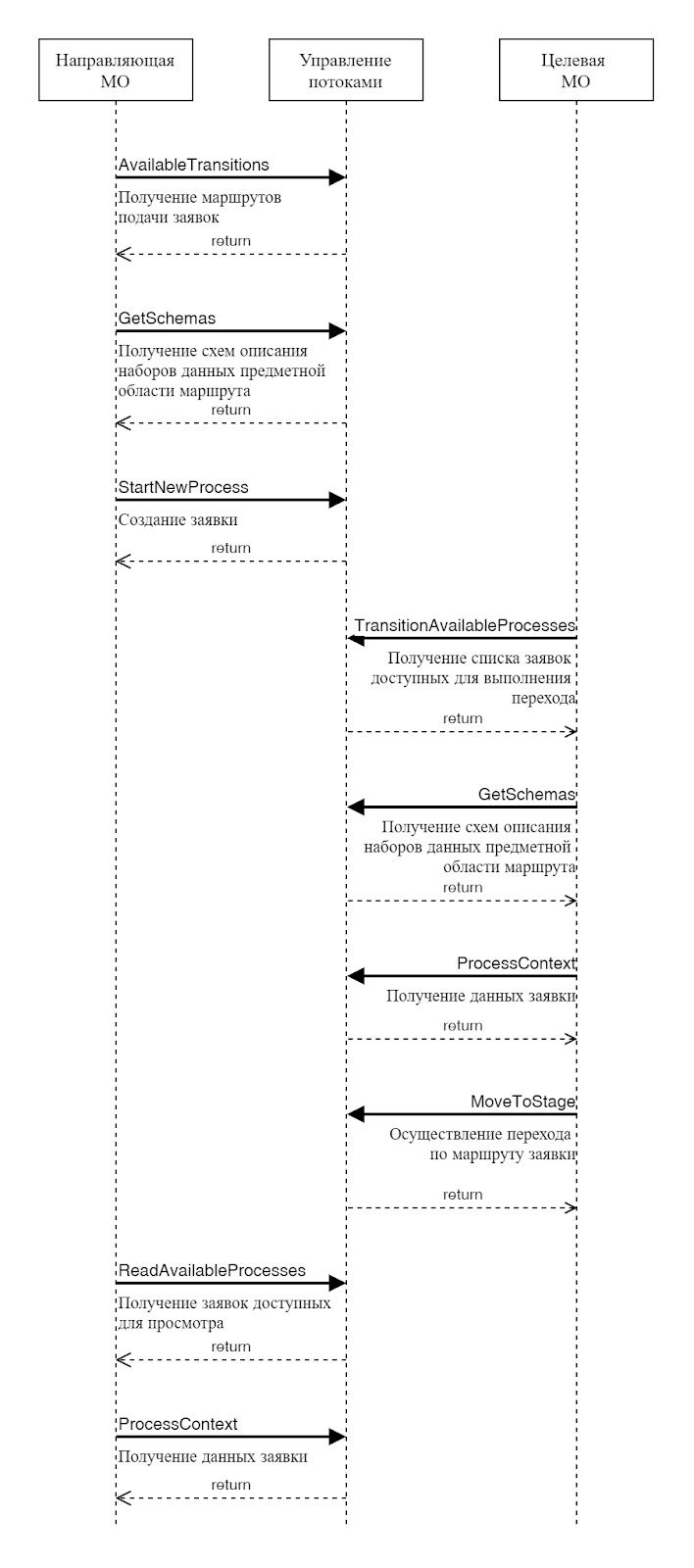


Рис. 3. Базовый сценарий работы с ТМ.РЕГИЗ.

# **Интеграция с иными сервисами ГИС РЕГИЗ в рамках работы с телемедицинскими заявками**

В ходе работы с телемедицинской заявкой врачи могут обмениваться информацией о сведениях, зарегистрированных в подсистемах ГИС РЕГИЗ:

1. Случаях медицинского обслуживания из ИЭМК;
2. Медицинских документах из ИЭМК;
3. Направлениях на медицинское обслуживание из УО;
4. Результат лабораторного исследования из ОДЛИ;
5. Результат инструментального исследования из ОДИИ;
6. Идентификатор карточки пациента из подсистемы Индекс пациента.

Эти сведения в рамках работы сервиса должны записываться и храниться в данных заявки. Предаются эти сведения с помощью методов создания заявки StartNewProcess и передвижения заявки по маршруту MoveToStage.

Соответственно формат передачи данных в контексте осуществления transition описывается в схеме передаваемых данных:

* описание transition следует получать, используя метод получения списка доступных переходов для движения заявки по маршруту GetAvailableNextStages или метод  получения доступных переходов для создания заявки AvailableTransitions;
* описание сопутствующей переходу схемы данных следует получать с помощью метода получения схемы по её идентификатору GetSchema/[id схемы]. Идентификатор схемы соответственно содержится в описании transition.

Таким образом МО могут передавать в рамках заявки на ТМК сведения об информационных объектах, хранящихся в подсистемах ГИС РЕГИЗ, в качестве ссылки (идентификатора объекта), что снимает необходимость повторно вносить подобную информацию вручную.

Регламенты информационного взаимодействия для получения данных из подсистем ГИС РЕГИЗ для работы с ТМ.РЕГИЗ не имеют отдельной специфики. Актуальные регламенты информационного взаимодействия размещены на официальном сайте СПб ГБУЗ МИАЦ - <https://spbmiac.ru/ehlektronnoe-zdravookhranenie/integracionnye-profili/>.

# **Описание методов обмена сведениями**

# **Получение списка переходов для создания заявки доступных пользователю (GetAvailableTransitions)**

**Описание параметров запроса**

Метод предоставляет перечень маршрутов и описания операций в рамках которых пользователь имеет право создавать заявки.

**Запрос**

Описание:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | roleContext | string | 1..1 | Значение ролевого контекста пользователя |
| 2 | metaDataFilter | string | 1..1 | Значение фильтра |

Пример:

|  |
| --- |
| POST http://[hostname]/api/Fhir/GetAvailableTransitions  authorization: N3[пробел][GUID передающей системы]  content-type: application/json  {    "resourceType" : "Parameters",    "parameter" : [{      "name" : "roleContext",      "part" : [{          "name":"snils",          "valueString":"123763187"      }],{      "name" : "metaDataFilter",      "part" : [{          "name":"org",          "valueString":"03284832468246812"      }]    }]  } |

**Ответ**

Описание структуры bundle. Структура используется для передачи набора структуры, описывающей характеристики ответа (OperationOutcome) по запросу, и структур, описывающих маршруты (PlanDifinition) и доступные пользователю операции для создания.

Описание структуры OperationOutcome:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | code | string | 1..1 | Код результата операции |
| 2 | diagnostics | string | 1..1 | Описание реузльтата операции |

Описание структуры PlanDifinition:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | identifier | backboneElement | 1..1 | Идентификатор маршрута |
| 2 | identifier.value | string | 1..1 | Значение идентификатора маршрута |
| 3 | usage | string | 0..1 | Метаданные маршрута |
| 4 | goal | backboneElement | 1…\* | Переходы |
| 5 | goal.id | string | 1..1 | Идентификатор перехода |
| 6 | goal.description.coding.code | string | 1..1 | Название перехода |
| 7 | goal.start.coding.code | string | 0..1 | Стартовый статус. Код справочника. |
| 8 | goal.target.detailCodeableConcept.coding.code | string | 1..1 | Конечный статус. Код справочника. |

Пример:

|  |
| --- |
| {  "resourceType": "Bundle",  "type": "transaction-response",  "entry": [    {      "resource": {        "resourceType": "OperationOutcome",        "issue": [          {            "severity": "information",            "code": "informational",            "diagnostics": "0"          }        ]      }    },    {      "resource": {        "resourceType": "PlanDefinition",        "identifier" : {           "value" : "7393b278-3a35-49e7-aeae-baefd66cca0c"        },        "status": "active",        "usage" : "84e05b4c-624f-4bd0-824f-3252b910215b",       "goal": [          {            "id" : "83ab6eba-a6eb-45a0-a0ac-a9c81a701ea6",            "description" : {              "coding": [                {                  "code": "Создать заявку"                }              ]            },            "start": {              "coding": [                {                  "code": null                }              ]            },            "target": [                 {                     "detailCodeableConcept": {                         "coding": [                             {                                 "code": "f518c380-2d6f-4cd2-9a6c-1fe6d16387fa"                             }                         ]                     }                 }             ]          }          ]      }    }  ]  } |

# **Создание заявки по маршруту (StartNewProcess)**

**Описание параметров запроса**

Метод создаёт заявку по маршруту согласно переданному переходу.

**Запрос**

Описание:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | workflowId | string | 1..1 | Идентификатор маршрута |
| 2 | name | string | 1..1 | Название заявки |
| 3 | initionalTransitionId | string | 1..1 | Идентификатор перехода для создания заявки |
| 4 | processContext | string | 1..1 | Набор данных |
| 5 | roleContext | string | 1..1 | Значение ролевого контекста пользователя |

Пример:

|  |
| --- |
| POST http://[hostname]/api/Fhir/StartNewProcess  authorization: N3[пробел][GUID передающей системы]  content-type: application/json  {    "resourceType" : "Parameters",    "parameter" : [     {      "name" : "workflowId",      "valueString" : "32c67fe5-3104-43b9-8ce9-34a042f187ae"    },    {      "name" : "name",      "valueUrl" : "Sz-process6-Fhir"    },    {      "name" : "initionalTransitionId",      "valueUrl" : "294eadb1-f52a-4f38-81ca-5d1a17f15168"    },    {      "name" : "processContext",      "part" : [{          "name":"name",          "valueString":"Петр"      },{          "name":"surname",          "valueString":"Сидоров"      }]    },    {      "name" : "roleContext",      "part" : [{          "name":"roleId",          "valueString":"7"      }]    }    ]  } |

**Ответ**

Описание структуры bundle. Структура используется для передачи набора структуры, описывающей характеристики ответа (OperationOutcome) по запросу, и структур, описывающих маршруты (Task) и доступные пользователю операции для создания.

Описание структуры OperationOutcome:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | code | string | 1..1 | Код результата операции |
| 2 | diagnostics | string | 1..1 | Описание реузльтата операции |

Описание структуры Task:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | identifier | backboneElement | 1..1 | Идентификатор заявки |
| 2 | identifier.value | string | 1..1 | Значение идентификатора заявки |
| 3 | basedOn | backboneElement | 1…\* | Ссылка на маршрут на основе которого создана заявка |
| 4 | basedOn.identifier.value | string | 1..1 | Идентификатор маршрута |
| 5 | businessStatus | сodeableConcept | 1..1 | Текущий статус заявки |
| 6 | businessStatus.coding.code | string | 1..1 | Идентификатор текущего статуса |

Пример:

|  |
| --- |
| {    "resourceType": "Bundle",    "type": "transaction-response",    "entry": [      {        "resource": {          "resourceType": "OperationOutcome",          "issue": [            {              "severity": "information",              "code": "informational",              "diagnostics": "0"            }          ]        }      },      {        "resource": {          "resourceType": "Task",          "identifier": [            {              "value": "dd2e3aa4-f0a8-4025-b880-6d988a6d9efd"            }          ],          "basedOn": [            {              "identifier": {                "value": "32c67fe5-3104-43b9-8ce9-34a042f187ae"              }            }          ],          "status": "in-progress",          "businessStatus": {            "coding": [              {                "code": "adadb6e8-4e33-495b-b813-20beca302651"              }            ]          }        }      }    ]  } |

# **Получение списка доступных для действия заявок (GetTransitionAvailableProcesses)**

**Описание параметров запроса**

Метод предназначен для получения списка заявок в статусах, которые имеют операции доступные для использования данному пользователю.

**Запрос**

Описание:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | roleContext | string | 1..1 | Значение ролевого контекста пользователя |
| 2 | workflowFilter | string | 0...1 | Параметры фильтра по схеме описания метаданных маршрута |
| 3 | processFilter | string | 0...1 | Параметры фильтра по схеме описания метаданных заявки |

Пример:

|  |
| --- |
| POST http://[hostname]/api/Fhir/GetTransitionAvailableProcesses  authorization: N3[пробел][GUID передающей системы]  content-type: application/json  {    "resourceType" : "Parameters",    "parameter" : [{      "name" : "roleContext",      "part" : [{          "name":"snils",          "valueString":"123763187"      }]    },{      "name" : "workflowFilter",      "part" : [{          "name":"org",          "valueString":"4cf9baf8-c930-4487-ae70-444403c4b421"      }]    },{      "name" : "processFilter",      "part" : [{          "name":"type",          "valueString":"3"      }]    }]  } |

**Ответ**

Описание структуры bundle. Структура используется для передачи набора структуры, описывающей характеристики ответа (OperationOutcome) по запросу, и структур, описывающих заявки (Task) и доступные пользователю операции для действия.

Описание структуры OperationOutcome:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | code | string | 1..1 | Код результата операции |
| 2 | diagnostics | string | 1..1 | Описание реузльтата операции |

Описание структуры Task:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | identifier | backboneElement | 1..1 | Идентификатор заявки |
| 2 | identifier.value | string | 1..1 | Значение идентификатора заявки |
| 3 | basedOn | backboneElement | 1…\* | Ссылка на маршрут на основе которого создана заявка |
| 4 | basedOn.identifier.value | string | 1..1 | Идентификатор маршрута |
| 5 | businessStatus | сodeableConcept | 1..1 | Текущий статус заявки |
| 6 | businessStatus.coding.code | string | 1..1 | Идентификатор текущего статуса |

Пример:

|  |
| --- |
| {   "resourceType": "Bundle",   "type": "transaction-response",   "entry": [     {       "resource": {         "resourceType": "OperationOutcome",         "issue": [           {             "severity": "information",             "code": "informational",             "diagnostics": "0"           }         ]       }     },     {       "resource": {         "resourceType": "Task",         "identifier": {           "value":"868a196f-14c8-4392-b417-876969a4c605"         },         "basedOn": [             {                 "identifier" : {                     "value": "7393b278-3a35-49e7-aeae-baefd66cca0c"                 }             }         ],         "status": "active",         "businessStatus" : {             "coding" : [                 {                   "code" : "4cf9baf8-c930-4487-ae70-444403c4b421"                 }             ]         }       }     }   ]  } |

# **Получение списка доступных для просмотра заявок (GetReadAvailableProcesses)**

**Описание параметров запроса**

Метод предназначен для получения списка заявок в статусах, которые доступны данному пользователю для просмотра и действия.

**Запрос**

Описание:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | roleContext | string | 1..1 | Значение ролевого контекста пользователя |
| 2 | filter | string | 0...1 | Параметры фильтра по схеме описания контекста маршрута и заявки |

Пример:

|  |
| --- |
| POST http://[hostname]/api/Fhir/GetReadAvailableProcesses  authorization: N3[пробел][GUID передающей системы]  content-type: application/json  {    "resourceType" : "Parameters",    "parameter" : [{      "name" : "roleContext",      "part" : [{          "name":"snils",          "valueString":"123763187"      }]    },{      "name" : "workflowFilter",      "part" : [{          "name":"org",          "valueString":"4cf9baf8-c930-4487-ae70-444403c4b421"      }]    },{      "name" : "processFilter",      "part" : [{          "name":"type",          "valueString":"3"      }]    }]  } |

**Ответ**

Описание структуры bundle. Структура используется для передачи набора структуры, описывающей характеристики ответа (OperationOutcome) по запросу, и структур, описывающих заявки (Task) и доступные пользователю для просмотра.

Описание структуры OperationOutcome:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | code | string | 1..1 | Код результата операции |
| 2 | diagnostics | string | 1..1 | Описание реузльтата операции |

Описание структуры Task:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | identifier | backboneElement | 1..1 | Идентификатор заявки |
| 2 | identifier.value | string | 1..1 | Значение идентификатора заявки |
| 3 | basedOn | backboneElement | 1…\* | Ссылка на маршрут на основе которого создана заявка |
| 4 | basedOn.identifier.value | string | 1..1 | Идентификатор маршрута |
| 5 | businessStatus | сodeableConcept | 1..1 | Текущий статус заявки |
| 6 | businessStatus.coding.code | string | 1..1 | Идентификатор текущего статуса |

Пример:

|  |
| --- |
| {   "resourceType": "Bundle",   "type": "transaction-response",   "entry": [     {       "resource": {         "resourceType": "OperationOutcome",         "issue": [           {             "severity": "information",             "code": "informational",             "diagnostics": "0"           }         ]       }     },     {       "resource": {         "resourceType": "Task",         "identifier": {           "value":"868a196f-14c8-4392-b417-876969a4c605"         },         "basedOn": [             {                 "identifier" : {                     "value": "7393b278-3a35-49e7-aeae-baefd66cca0c"                 }             }         ],         "status": "active",         "businessStatus" : {             "coding" : [                 {                   "code" : "4cf9baf8-c930-4487-ae70-444403c4b421"                 }             ]         }       }     }   ]  } |

# **Получение заявки по идентификатору**

**Описание параметров запроса**

Метод возвращает данные о заявке по её идентификатору.

**Запрос**

Пример:

|  |
| --- |
| POST http://[hostname]/api/Process/[id заявки]  authorization: N3[пробел][GUID передающей системы]  content-type: application/json |

**Ответ**

Описание структуры bundle. Структура используется для передачи набора структуры, описывающей характеристики ответа (OperationOutcome) по запросу, и структуры, описывающей характеристики задачи (Task).

Описание структуры OperationOutcome:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | code | string | 1..1 | Код результата операции |
| 2 | diagnostics | string | 1..1 | Описание реузльтата операции |

Описание структуры Task:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | identifier | backboneElement | 1..1 | Идентификатор заявки |
| 2 | identifier.value | string | 1..1 | Значение идентификатора заявки |
| 3 | basedOn | backboneElement | 1…\* | Ссылка на маршрут на основе которого создана заявка |
| 4 | basedOn.identifier.value | string | 1..1 | Идентификатор маршрута |
| 5 | businessStatus | сodeableConcept | 1..1 | Текущий статус заявки |
| 6 | businessStatus.coding.code | string | 1..1 | Идентификатор текущего статуса |
| 7 | description | string | 1..\* | Заголовок заявки |
| 8 | authoredOn | dateTime | 1..1 | Время создания заявки |
| 9 | lastModified | dateTime | 1..1 | Время последнего изменения |
| 10 | input | backboneElement | 1..1 | Объект контекста заявки |
| 11 | input.type.coding.valueId | id | 1..1 | Идентификатор объекта контекста заявки |

Пример:

|  |
| --- |
| {    "resourceType": "Bundle",    "type": "transaction-response",    "entry": [      {        "resource": {          "resourceType": "OperationOutcome",          "issue": [            {              "severity": "information",              "code": "informational",              "diagnostics": "0"            }          ]        }      },      {        "resource": {          "resourceType": "Task",          "identifier": [            {              "value": "dd2e3aa4-f0a8-4025-b880-6d988a6d9efd"            }          ],          "basedOn": [            {              "identifier": {                "value": "32c67fe5-3104-43b9-8ce9-34a042f187ae"              }            }          ],          "status": "in-progress",          "businessStatus": {            "coding": [              {                "code": "adadb6e8-4e33-495b-b813-20beca302651"              }            ]          },          "description": "Sz-process6-Fhir",          "authoredOn": "02.04.2020 12:34:05 +03:00",          "lastModified": "02.04.2020 12:34:05 +03:00",          "input": [            {              "type": {                "coding": [                  {                    "code": "3"                  }                ]              },              "valueId": "76843a46-e14b-49e0-b4b4-1b14c56363c3"            }          ]        }      }    ]  } |

# **Получение объекта контекста процесса (GetProcessContext)**

**Описание параметров запроса**

Метод предназначен для получения набора данных (объекта контекста), собранного при работе с заявкой.

**Запрос**

Пример:

|  |
| --- |
| POST http://[hostname]/api/Fhir/GetProcessContext?processId=[id заявки]  authorization: N3[пробел][GUID передающей системы]  content-type: application/json |

**Ответ**

Описание структуры OperationOutcome. Структура используется для передачи набора структуры, описывающей характеристики ответа по запросу и структурированного набора данных заявки.

Описание структуры OperationOutcome:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | code | string | 1..1 | Код результата операции |
| 2 | diagnostics | string | 1..1 | Описание реузльтата операции |
| 3 | expression | string | 0..\* | Объект контекста структурированный по схеме предметной области |

Пример:

|  |
| --- |
| {   "resourceType": "OperationOutcome",   "issue": [     {        "severity": "information",        "code": "informational",        "diagnostics": "0",        "expression": "{        \"required\": [          \"snils\",          \"patientId\"        ],        \"properties\": {          \"snils\": {            \"type\": \"string\"          },          \"patientId\": {            \"type\": \"string\"          }        }      }"     }   ]  } |

# **Получение идентификаторов следующих возможных состояний для процесса (GetAvailableNextStages)**

**Описание параметров запроса**

Метод предоставляет перечень переходов, которые доступны пользователю в текущем состоянии заявки.

**Запрос**

Описание:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | roleContext | string | 1..1 | Значение ролевого контекста пользователя |

Пример:

|  |
| --- |
| POST http://[hostname]/api/Fhir/GetAvailableNextStages?processId=[id заявки]  authorization: N3[пробел][GUID передающей системы]  content-type: application/json |

**Ответ**

Описание структуры bundle. Структура используется для передачи набора структуры, описывающей характеристики ответа (OperationOutcome) по запросу, и структур, описывающих маршруты (PlanDifinition) и доступные пользователю операции для перехода в данном состоянии заявки.

Описание структуры OperationOutcome:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | code | string | 1..1 | Код результата операции |
| 2 | diagnostics | string | 1..1 | Описание реузльтата операции |

Описание структуры PlanDifinition:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | identifier | backboneElement | 1..1 | Идентификатор маршрута |
| 2 | identifier.value | string | 1..1 | Значение идентификатора маршрута |
| 3 | goal | backboneElement | 1…\* | Переходы |
| 4 | goal.id | string | 1..1 | Идентификатор перехода |

Пример:

|  |
| --- |
| {  "resourceType": "Bundle",  "type": "transaction-response",  "entry": [    {      "resource": {        "resourceType": "OperationOutcome",        "issue": [          {            "severity": "information",            "code": "informational",            "diagnostics": "0"          }        ]      }    },    {      "resource": {        "resourceType": "PlanDefinition",        "identifier" : {           "value" : "32c67fe5-3104-43b9-8ce9-34a042f187ae"        },        "status": "active",        "goal": [          {            "id" : "4f329dae-ee5d-4e32-9753-80c25e24abd7"          }        ]      }    }  ]  } |

# **Получение описания правил перехода по его идентификатору (GetTransition)**

**Описание параметров запроса**

Метод возвращает описание условий перехода к новому состоянию заявки.

**Запрос**

Пример:

|  |
| --- |
| POST http://[hostname]/api/Queries/GetTransition/[id перехода]  authorization: N3[пробел][GUID передающей системы]  content-type: application/json |

**Ответ**

Описание структуры bundle. Структура используется для передачи набора структуры, описывающей характеристики ответа (OperationOutcome) по запросу, и структуры, описывающей характеристики задачи (PlanDifinition).

Описание структуры OperationOutcome:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | code | string | 1..1 | Код результата операции |
| 2 | diagnostics | string | 1..1 | Описание реузльтата операции |

Описание структуры PlanDifinition:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | identifier | backboneElement | 1..1 | Идентификатор маршрута |
| 2 | identifier.value | string | 1..1 | Значение идентификатора маршрута |
| 3 | goal | backboneElement | 1…\* | Переходы |
| 4 | goal.id | string | 1..1 | Идентификатор перехода |
| 5 | goal.description.coding.code | string | 1..1 | Название перехода |
| 6 | goal.start.coding.code | string | 0..1 | Стартовый статус. Код справочника. |
| 7 | goal.documentation.id | string | 0..1 | Идентификатор схемы описывающей входящий набор данных |
| 8 | goal.target.detailCodeableConcept.coding.code | string | 1..1 | Конечный статус. Код справочника. |
| 9 | action | backboneElement | 0..1 | Проверки и вызываемые операции при совершении перехода к новому состоянию заявки |
| 10 | action.goalId | string | 1..1 | Идентификатор перехода |
| 11 | action.description | string | 1..1 | “validatorsIds” - проверка  “callbacksIds” - уведомление |
| 12 | action.dynamicValue.path | string | 1..1 | Идентификатор проверки или уведомления |

Пример:

|  |
| --- |
| {  "resourceType": "Bundle",  "type": "transaction-response",  "entry": [    {      "resource": {        "resourceType": "OperationOutcome",        "issue": [          {            "severity": "information",            "code": "informational",            "diagnostics": "0"          }        ]      }    },    {      "resource": {        "resourceType": "PlanDefinition",        "status": "active",        "goal": [          {            "identifier" : {               "value" : "83ab6eba-a6eb-45a0-a0ac-a9c81a701ea6"            },            "status": "active",            "description" : {              "coding": [                {                  "code": "Согласовать"                }              ]            },            "start": {              "coding": [                {                  "code": "92faae41-e895-44ba-9618-4e25ea4e27ec"                }              ]            },             "documentation" : [                 {                 "id":"d11e053c-417e-406e-94db-bd170132ca00"                 }             ],            "target": [                 {                     "detailCodeableConcept": {                         "coding": [                             {                                 "code": "f518c380-2d6f-4cd2-9a6c-1fe6d16387fa"                             }                         ]                     }                 }             ]          }          ],          "action": [           {             "goalId": "83ab6eba-a6eb-45a0-a0ac-a9c81a701ea6",             "description": "validatorIds",             "dynamicValue": [               {                 "path": "44749795-e68c-4af7-b7c2-e4deced81564"               }             ]           },           {             "goalId": "83ab6eba-a6eb-45a0-a0ac-a9c81a701ea6",             "description": "callbackIds",             "dynamicValue": [               {                 "path": "d11e053c-417e-406e-94db-bd170132ca00"               }             ]           }         ]      }    }  ]  } |

# **Получение схемы данных передаваемых при осуществлении перехода (GetSchema)**

**Описание параметров запроса**

Метод предназначен для получения схемы данных (объекта контекста), необходимого для осуществления перехода к новому состоянию заявки.

**Запрос**

Пример:

|  |
| --- |
| POST http://[hostname]/api/Queries/GetSchema/[id схемы]  authorization: N3[пробел][GUID передающей системы]  content-type: application/json |

**Ответ**

Описание структуры bundle. Структура используется для передачи набора структуры, описывающей характеристики ответа (OperationOutcome) по запросу, и структур, описывающих схему данных (StructureDifinition) необходимых для осуществления перехода.

В описании структур, передаваемых данных в формате json-схемы, содержится описания типов передаваемых данных, кратности и обязательности.

Например, формат передачи справочных значений по формату HL7 FHIR:

|  |
| --- |
| "CodeableConcept": {   "description": "A concept that may be defined by a formal reference to a terminology or ontology or may be provided by text.",   "properties": {     "id": {       "description": "Unique id for the element within a resource (for internal references). This may be any string value that does not contain spaces.",       "$ref": "#/definitions/string"     },     "extension": {       "description": "May be used to represent additional information that is not part of the basic definition of the element. To make the use of extensions safe and manageable, there is a strict set of governance  applied to the definition and use of extensions. Though any implementer can define an extension, there is a set of requirements that SHALL be met as part of the definition of the extension.",       "items": {         "$ref": "#/definitions/Extension"       },       "type": "array"     },     "coding": {       "description": "A reference to a code defined by a terminology system.",       "items": {         "$ref": "#/definitions/Coding"       },       "type": "array"     },     "text": {       "description": "A human language representation of the concept as seen/selected/uttered by the user who entered the data and/or which represents the intended meaning of the user.",       "$ref": "#/definitions/string"     },     "\_text": {       "description": "Extensions for text",       "$ref": "#/definitions/Element"     }   },   "additionalProperties": false  } |

Справочные материалы в обмене данными в заявках используются в рамках справочников НСИ. В случае использования справочника, в описании структуры данных "description" указывается код справочника.

Если в рамках заявки требуется передать информацию о пациенте, зарегистрированном в ГИС РЕГИЗ, случае ИЭМК, результатах лабораторных или инструментальных исследований, направлении на оказание медицинской помощи, то согласно схеме описания данных для осуществления операции над заявкой потребуется указать идентификаторы данных объектов из соответствующих сервисов РЕГИЗ: Индекс пациента, ИЭМК, ОДЛИ, ОДИИ, УО.

Актуальные регламенты информационного взаимодействия размещены на официальном сайте СПб ГБУЗ МИАЦ - <https://spbmiac.ru/ehlektronnoe-zdravookhranenie/integracionnye-profili/>.

Соответственно в качестве примера передача ссылки на карточку пациента сервиса “Индекс пациента” в схеме описания передачи данных:

|  |
| --- |
| "subject": {    "description": "Пациент",    "required": ["reference"],    "properties": {      "type": "object",      "reference": {        "description": "Идентификатор пациента в MPI",        "type": "string"     }   }  } |

Пример передачи данных об идентификаторе пациента сервиса “Индекс пациента”:

|  |
| --- |
| "subject": {      "reference": "a7ad714e-7c68-4950-ac7d-408bb68e23e9"  } |

#### **Получение идентификаторов случаев медицинского обслуживания и медицинских документов с помощью виджета Портала врача**

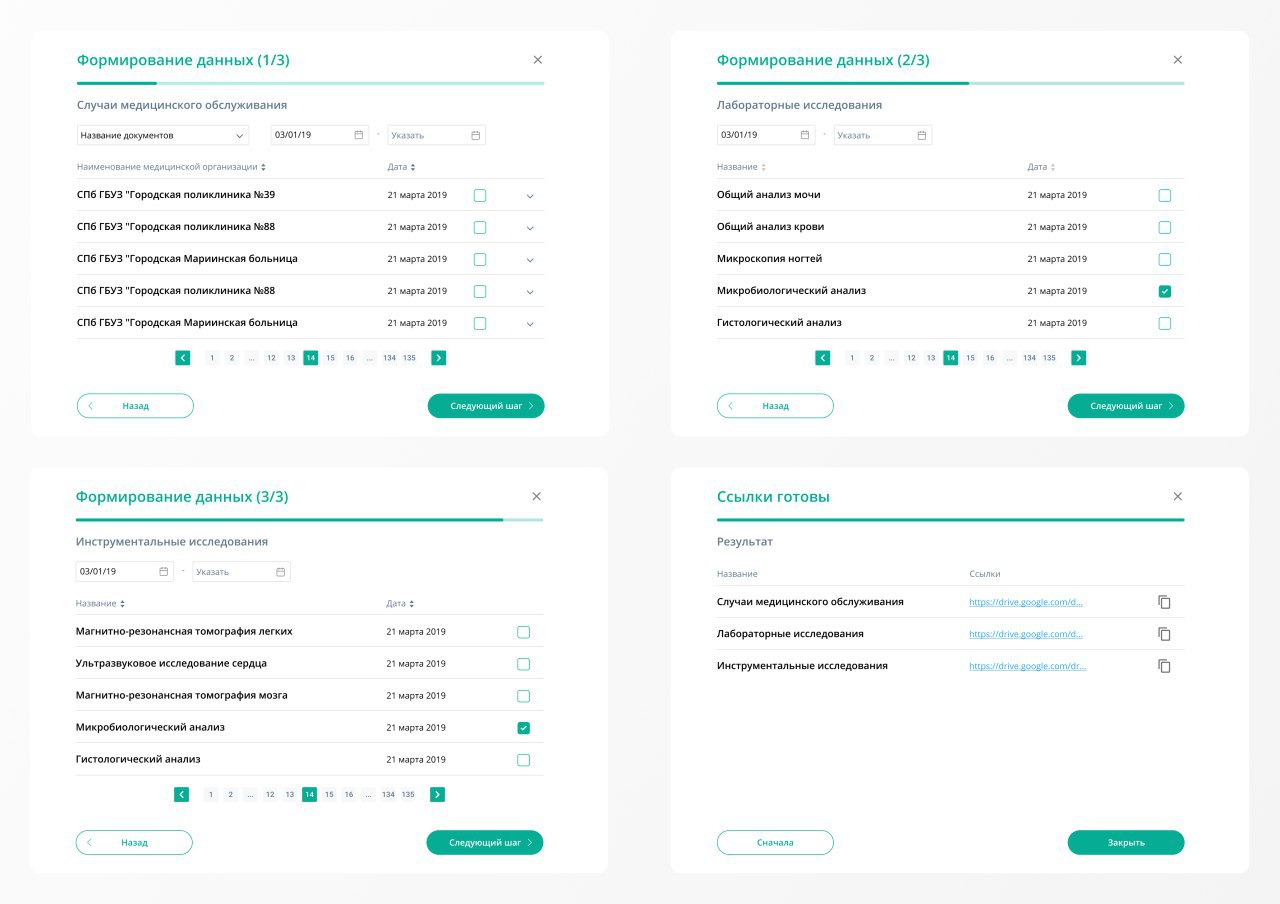
Также для формирования ссылки на случаи медицинского обслуживания можно воспользоваться виджетом Портала врача:

[portal]\wizard\{idPatient}

Порядок авторизации аналогичен главной форме Портала врача.

Мастер состоит из 4-х страниц:

1. Выбор случая медицинского обслуживания.
2. Выбор лабораторных исследований.
3. Выбор инструментальных исследований.
4. Страница готовых ссылок.



Общими для первых 3-х страниц являются элементы:

* Персональные данные пациента - аналогично главной форме Портала.
* Выбор периода - два выпадающих списка, позволяющих задать период отображения данных.  
  По умолчанию началом периода является первый день текущей недели, окончание периода не задано (предполагается текущая дата).
* Пагинация - перемещение по постраничной выдаче результата в количестве 5 элементов.
* Кнопки навигации “Следующий шаг“, ”Назад” - перемещение по страницам мастера.

Отображается список ссылок, сформированный на предыдущих шагах.

Для каждой ссылки отображается ее текстовое представление и кнопка копирования содержания в буфер обмена.

Описание структуры OperationOutcome:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | contained | DomainResource | 0..\* | Содержит структуру “parameter” и поля type, areaId, schema |
| 2 | contained.resource.parameter.type | string | 1..1 | Тип схемы данных. Согласно справочнику.  0: основная схема  1: метаданные маршрута  2: метаданные заявки  3: набор данных для перехода |
| 3 | contained.resource.parameter.areaId | string | 1..1 | Идентификатор предметной области |
| 4 | contained.resource.parameter.schema | string | 1..1 | JSON-схема |
| 5 | url | uri | 0..\* | Идентификатор схемы |
| 6 | name | string | 0..\* | Название схемы |
| 7 | experimental | boolean | 1..1 | Является заблокированной |

Пример:

|  |
| --- |
| {  "resourceType": "Bundle",  "type": "transaction-response",  "entry": [    {      "resource": {        "resourceType": "OperationOutcome",        "issue": [          {            "severity": "information",            "code": "informational",            "diagnostics": "0"          }        ]      }    },    {      "resource": {        "resourceType": "StructureDefinition",        "contained":[            {                 "resource": {                     "resourceType" : "Parameters",                       "parameter" : [{                         "name" : "type",                         "valueString" : "3"                       },{                         "name" : "areaId",                         "valueString" : "868a196f-14c8-4392-b417-876969a4c605"                       },{                         "name" : "schema",                         "valueString" : "{                           \"required\": [                           \"snils\",                           \"patientId\"                           ],                           \"properties\": {                             \"snils\": {                               \"type\": \"string\"                           },                           \"patientId\": {                             \"type\": \"string\"                          }                       }                   }"                       }]                 }            }        ],        "url": "d11e053c-417e-406e-94db-bd170132ca00",        "name": "Переход",         "experimental" : false      }    }  ]  } |

# **Осуществление перехода в указанное состояние в рамках указанного процесса (MoveToStage)**

**Описание параметров запроса**

Метод позволяет осуществить запрос перехода заявки в новое состояние согласно указанному правилу перехода.

**Запрос**

Описание:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | processId | string | 1..1 | Идентификатор заявки |
| 2 | transitionId | string | 1..1 | Идентификатор перехода |
| 3 | processContext | string | 1..1 | Набор данных |
| 4 | roleContext | string | 1..1 | Значение ролевого контекста пользователя |

Пример:

|  |
| --- |
| POST http://[hostname]/api/Fhir/MoveToStage  authorization: N3[пробел][GUID передающей системы]  content-type: application/json  {    "resourceType" : "Parameters",    "parameter" : [{      "name" : "processId",      "valueString" : "b894cd31-76be-4269-9e0b-0f2eab7d3e6e"    },{      "name" : "transitionId",      "valueUrl" : "58de33e6-74f9-4185-8bb7-0ff2ecbc35a2"    },{      "name" : "processContext",      "part" : [{          "name":"name",          "valueString":"Петр"      },{          "name":"surname",          "valueString":"Сидоров"      }]    },{      "name" : "roleContext",      "part" : [{          "name":"snils",          "valueString":"123763187"      }]    }]  } |

**Ответ**

Описание структуры bundle. Структура используется для передачи набора структуры, описывающей характеристики ответа (OperationOutcome) по запросу, и структуры, описывающей характеристики задачи (Task).

Описание структуры OperationOutcome:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | code | string | 1..1 | Код результата операции |
| 2 | diagnostics | string | 1..1 | Описание реузльтата операции |

Описание структуры Task:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | identifier | backboneElement | 1..1 | Идентификатор заявки |
| 2 | identifier.value | string | 1..1 | Значение идентификатора заявки |
| 3 | basedOn | backboneElement | 1…\* | Ссылка на маршрут на основе которого создана заявка |
| 4 | basedOn.identifier.value | string | 1..1 | Идентификатор маршрута |
| 5 | businessStatus | сodeableConcept | 1..1 | Текущий статус заявки |
| 6 | businessStatus.coding.code | string | 1..1 | Идентификатор текущего статуса |
| 7 | description | string | 1..\* | Заголовок заявки |
| 8 | authoredOn | dateTime | 1..1 | Время создания заявки |
| 9 | lastModified | dateTime | 1..1 | Время последнего изменения |
| 10 | input | backboneElement | 1..1 | Объект контекста заявки |
| 11 | input.type.coding.valueId | id | 1..1 | Идентификатор объекта контекста заявки |

Пример:

|  |
| --- |
| {   "resourceType": "Bundle",   "type": "transaction-response",   "entry": [     {       "resource": {         "resourceType": "OperationOutcome",         "issue": [           {             "severity": "information",             "code": "informational",             "diagnostics": "0"           }         ]       }     },     {       "resource": {         "resourceType": "Task",         "identifier": {           "value":"58de33e6-74f9-4185-8bb7-0ff2ecbc35a2"         },         "basedOn": [             {                 "identifier" : {                     "value": "b894cd31-76be-4269-9e0b-0f2eab7d3e6e"                 }             }         ],         "status": "active",         "businessStatus" : {             "coding" : [                 {                   "code" : "e616b6f9-837e-45c6-910c-6e84daac0eb1"                 }             ]         },         "description" : "Кардиология. Петров О. И.",         "authoredOn" : "<1/2/2000 12:00:00 AM",         "lastModified" : "1/2/2000 13:12:00 AM",         "input" : [{             "type": {                 "coding" : [                     {                       "code" : "3"                     }                 ]             },             "valueId" : "4cf9baf8-c930-4487-ae70-444403c4b421"         }]       }     }   ]  } |

# **Коды возвращаемых ошибок**

|  |  |
| --- | --- |
| Код | Описание |
| 0 | Ошибок не найдено |
| 1 | Внутренняя ошибка приложения |
| 2 | Ошибка валидации выполнения операции. Неверный код передаваемой сущности, отсутствие обязательных данных согласно спецификациям |
| 3 | Множественный переход |
| 11 | Указанный маршрут не найден |
| 12 | Указанного статуса не существует |
| 14 | Указанного валидатора не существует |
| 15 | Указанного отклика не существует |
| 16 | Заявка не найдена |
| 17 | Указанной предметной области не существует |
| 18 | Указанной схемы данных не существует |
| 19 | Указанной операции (transition) не существует |
| 32 | Нет данных ожидаемых для осуществления перехода или создания заявки |
| 33 | Нет метаданных описания маршрута |
| 42 | Данные расширения схемы не разрешены |
| 51 | Метаданные маршрута не найдены |
| 52 | Метаданные заявки не найдены |