**ГОСУДАРСТВЕННая ИНФОРМАЦИОННая**

**СИСТЕМа санкт-петербУрга «Региональный фрагмент Единой государственной информационной сиСТемы В СФЕРЕ ЗДРАВООХРАНЕНИЯ»**

**ОПИСАНИЕ ИНТЕГРАЦИОННЫХ ПРОФИЛЕЙ**

**ПОДСИСТЕМЫ «ТЕЛЕМЕДИЦИНА»**

На 72 листах

# **История версий**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Версия** | **Дата** | **Автор** | **Примечание** |
| 1 | 1.1 draft | 20.05.2020 | Туркин 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. |
| 2 | 1.2 | 06.09.2020 | Туркин А. | Добавлено описание метода получения ссылок на чаты и видеоконференцсвязь для участников заявок на ТМК. Добавлен контрольный пример. |
| 3 | 1.3 | 18.09.2020 | Туркин А. | Добавлено описание логики работы фильтрации в методах получения списков заявок и доступных для создания заявок маршрутов.Добавлено описание типовой структуры схемы ролевого контекста для пользователей МИС. |
| 4 | 1.4 | 24.09.2020 | Туркин А. | Добавлено описание метода получения маршрута обработки заявки (GetWorkflow). Обновлена схема базового сценария работы с ТМ.РЕГИЗ.Поправлены ошибки в типах данных. |
| 5 | 1.5 | 25.09.2020 | Туркин А. | Расшифровка параметров заявки контрольного примера разбита на части согласно каждому запросу.  |

Содержание

[**История версий** 2](#_Toc51856109)

[**Определения, обозначения и сокращения** 4](#_Toc51856110)

[**Общая информация** 5](#_Toc51856111)

[**Описание решения** 7](#_Toc51856112)

[**Внутренняя логика сервиса** 1](#_Toc51856113)

[**Базовый сценарий работы с сервисом** 4](#_Toc51856114)

[**Интеграция с иными сервисами ГИС РЕГИЗ в рамках работы с телемедицинскими заявками** 7](#_Toc51856115)

[**Описание методов обмена сведениями** 8](#_Toc51856116)

[**1.** **Получение списка переходов для создания заявки доступных пользователю (GetAvailableTransitions)** 8](#_Toc51856117)

[**2.** **Получение маршрута обработки заявки по идентификатору (GetWorkflow)** 11](#_Toc51856118)

[**3.** **Создание заявки по маршруту (StartNewProcess)** 15](#_Toc51856119)

[**4.** **Получение списка доступных для действия заявок (GetTransitionAvailableProcesses)** 24](#_Toc51856120)

[**5.** **Получение списка доступных для просмотра заявок (GetReadAvailableProcesses)** 30](#_Toc51856121)

[**6.** **Получение заявки по идентификатору** 36](#_Toc51856122)

[**7.** **Получение объекта контекста процесса (GetProcessContext)** 38](#_Toc51856123)

[**8.** **Получение идентификаторов следующих возможных состояний для процесса (GetProcessWithAvailableTransitions)** 40](#_Toc51856124)

[**9.** **Получение описания правил перехода по его идентификатору (GetTransition)** 43](#_Toc51856125)

[**10.** **Получение схемы данных передаваемых при осуществлении перехода (GetSchema)** 44](#_Toc51856126)

[**11.** **Осуществление перехода в указанное состояние в рамках указанного процесса (MoveToStage)** 49](#_Toc51856127)

[**12.** **Получение ссылок на чаты и видеоконференцсвязь для участников заявки** 54](#_Toc51856128)

[**Контрольный пример** 58](#_Toc51856129)

# **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.

Пример 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.

Для определения ролевого доступа к просмотру и работе с заявками используется «ролевой контекст». Для предметной области «Телемедицина» ролевой контекст определён набором json-схем. Основной ролевой схемой для работы пользователей МИС является данная схема:

{

 "type": "object",

 "$schema": "http://json-schema.org/draft-04/schema#",

 "required": [

 "Organization",

 "Role"

 ],

 "properties": {

 "Role": {

 "type": "string",

 "title": "Роль пользователя",

 "description": "Заполняется согласно справочнику НСИ 1.2.643.5.1.13.2.1.1.734"

 },

 "Organization": {

 "type": "string",

 "title": "Организация пользователя",

 "examples": [

 "Organization/a7ad714e-7c68-4950-ac7d-408bb68e23e9"

 ],

 "description": "Заполняется согласно справочнику НСИ 1.2.643.2.69.1.1.1.64"

 },

 "SNILS": {

 "type": "string",

 "title": "СНИЛС",

 "examples": [

 "89322342342"

 ],

 "description": "СНИЛС сотрудника заполняется без пробелов и тире"

 }

 }

}

Пример ролевого контекста согласно данной схеме:

{

 "Role": "DOCTOR",

 "Organization": "Organization/dd4a7bf9-f853-46e3-af27-11a52ca1178b",

 "SNILS": "89322342342"

}

Так же идентификатор схемы ролевого контекста возможно получить при получении сведений о маршруте (getWorkflow) и операции над заявкой (getTransition). Саму схему впоследствии можно запросить методом «getSchema».

# **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. Направляющая МО по выбранному маршруту запрашивает из системы описание условий перехода к новому состоянию заявки.
3. 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.
4. Направляющая МО создаёт заявку.
5. Целевая (консультирующая) МО запрашивает списки заявок, доступных для выполнения доступных операций согласно их текущему состоянию.
6. По полученным данным целевая МО запрашивает схемы передачи данных необходимых для осуществления операции изменения заявки.
7. Для принятия решения о последующих действиях целевая МО запрашивает содержание заявки.
8. 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.
9. Направляющая МО запрашивает список заявок, доступных для просмотра.
10. 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. Случаи медицинского обслуживания из ИЭМК;
2. Медицинские документы из ИЭМК;
3. Направления на медицинское обслуживание из УО;
4. Результаты лабораторного исследования из ОДЛИ;
5. Результаты 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 GetTransitionAvailableProcesses или метод  получения доступных переходов для создания заявки GetAvailableTransitions;
* 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 | object | 1..1 | Значение ролевого контекста пользователя |
| 2 | metaDataFilter | object | 1..1 | Значение фильтра |
| 3 | Skip | number | 0..1 | Количество пропущенных элементов в выборке |
| 4 | Take | number | 0..1 | Количество элементов в списке |

**Фильтр**

Объект для фильтрации по маршруту. Все поля являются опциональными (не обязательными).

Пример конструкции фильтра:

  "WorkflowFilter": {

    "name": "",

 "metadata": {},

  }

* Name - название маршрута (точное совпадение)
* Metadata - фильтр по объекту метаданных маршрута (правила сопоставления Json-объектов см. ниже)

Метаданные маршрута описываются по соответствующей схеме и указываются при его создании или редактировании администратором системы.

**Правила сопоставления Json-объектов в качестве фильтров метаданных маршрута или заявки:**

В качестве фильтра принимается Json-объект, в котором путь к искомому значению будет таким же, как в объекте, где производится поиск. Например, в метаданных процесса есть следующие данные:

 {

  "patientName": "Alexander",

  "diagnosis": {

    "name": "some name",

    "code": 315

  },

  "participant": {

    "name": "Ivan",

    "cpecialityCodes": [

      103,

      203,

      304

    ]

  }

}

Для доступа к полю "code" в объекте "diagnosis" нужно повторить путь к полю:

  {

  "diagnosis": {

    "code": 315

  }

}

Все выбранные данные в объекте в любом случае будут приведены к строке, и выбор сопоставление производится с помощью SQL-конструкции LIKE (поиск вхождения подстроки). Таким образом, следующее выражение будет эквивалентно предыдущему:

 {

  "diagnosis": {

    "code": "315"

  }

}

Для выбора нескольких значений, одного и того же поля можно использовать перечисление значений в массиве, однако в таком случае будет использовано полное совпадение:

 {

  "diagnosis": {

    "code": [

      314,

      315,

      317

    ]

  }

}

В случае если в объекте фильтра будет несколько полей — итоговое выражение будет собираться с учётом того, что все выражения должны быть истиной.

{

  "patientName": [

    "Alex",

    "Sam"

  ],

  "diagnosis": {

    "name": "some name",

    "code": [

      314,

      315,

      317

    ]

  }

}

Пример:

|  |
| --- |
| POST http://[hostname]/api/Queries/GetAvailableTransitionsauthorization: N3[пробел][GUID передающей системы] content-type: application/json{ "RoleContext": [], "MetaDataFilter": { "Organization": "dfe3eec2-8a79-4921-9b58-0ce03a5e6c10"  }, "Skip": 0, "Take": 20} |

**Ответ**

Описание структуры ответа:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | ErrorCode | string | 1..1 | Код результата операции |
| 2 | Message | string | 1..1 | Описание результата операции |
| 3 | WorkflowId | string | 1..1 | Идентификатор маршрута |
| 4 | WorkflowMetadata | object | 0..1 | Метаданные маршрута |
| 5 | TransitionIds | array | 0…\* | Массив идентификаторов переходов доступных для создания заявки по данному маршруту |

Пример:

|  |
| --- |
| { "result": [ { "workflowMetadata": { "Profile": "126", "Urgency": "3", "Organization": "dfe3eec2-8a79-4921-9b58-0ce03a5e6c10", "FHIRSupported": true }, "workflowId": "5c2c06cf-985b-4905-816e-bc0dec57dd9c", "transitionIds": [ "c0ea0245-6e72-4dba-b37c-21e4bdaffdad" ] } ], "success": true, "errorCode": 0, "message": null, "stackTrace": null} |

# **Получение маршрута обработки заявки по идентификатору (GetWorkflow)**

**Описание параметров запроса**

Метод возвращает данные о статусах и операциях на маршруте по идентификатору. С помощью возвращаемых данных можно узнать все возможные состояния заявки, перечень возможных операций и переходов между статусами, выполняемые проверки при попытках выполнения переходов и условия доступности заявок на каждом из статусов.

**Запрос**

Пример:

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

**Ответ**

Описание структуры ответа:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | ErrorCode | string | 1..1 | Код результата операции |
| 2 | Message | string | 1..1 | Описание результата операции |
| 3 | result.Id | string | 1..1 | Системный идентификатор маршрута |
| 4 | result.Name | string | 1…\* | Название маршрута |
| 5 | result.Description | string | 1..1 | Описание маршрута |
| 6 | result.IsContruction | string | 1..1 | Статус отладки. Если передано True, то маршрут является пилотным |
| 7 | result.AreaId | string | 1..1 | Идентификатор предметной области, в рамках которой существует маршрут |
| 8 | result.AreaName | string | 1..1 | Имя предметной области, в рамках которой существует маршрут |
| 9 | result.MetadataID | Array | 0..1 | Идентификатор метаданных маршрута |
| 10 | result.Metadata | object | 0..1 | Структура метаданных маршрута |
| 11 | result.Stages | Array | 0..\* | Статусы |
| 12 | result.Stages.StageId | object | 0..1 | Статус |
| 13 | result.Stages.StageId.Id | string | 1..1 | Идентификатор статуса |
| 14 | result.Stages.StageId.Name | string | 1..1 | Название статуса |
| 15 | result.Stages.StageId.Description | string | 0..1 | Описание статуса |
| 16 | result.Stages.StageId.Validators | Array | 0..1 | Проверки, выполняемые для доступа к данным по заявке, когда она находится в данном статусе |
| 17 | result.Stages.StageId.Validators.Id | String | 1..1 | Идентификатор проверки |
| 18 | result.Stages.StageId.Validators.Name | String | 1..1 | Название проверки |
| 19 | result.Stages.StageId.IsDisabled | boolean | 1..1 | Статус блокировки |
| 20 | result.Stages.StageId.BusinessStatus | Object | 0..1 | Статус согласно справочнику НСИ |
| 21 | result.Stages.StageId.BusinessStatus.System | string | 0..1 | OID справочника статусов |
| 22 | result.Stages.StageId.BusinessStatus.Code | string | 0..1 | Код статуса |
| 23 | result.Transitions | Array | 0..1 | Операции |
| 24 | result.Transitions.TransitionId | Object | 1..1 | Операция (переход) |
| 25 | result.Transitions.TransitionId.Id | String | 1..1 | Идентификатор операции (перехода) |
| 26 | result.Transitions.TransitionId.Name | String | 1..1 | Название операции (перехода) |
| 27 | result.Transitions.TransitionId.FromStageId | String | 0..1 | Идентификатор статуса заявки, в котором можно совершить операцию. Если передаётся «null», то операция используется для создания заявки |
| 28 | result.Transitions.TransitionId.ToStageId | String | 1..1 | Идентификатор статуса заявки, в котором она окажется после осуществления операции (перехода) |
| 29 | result.Transitions.TransitionId.SchemaId | String | 0..1 | Идентификатор схемы данных необходимых для выполнения операции |
| 30 | result.Transitions.TransitionId.Validators | Object | 1..\* | Проверки,выполняемые при попытке выполнении операции (перехода) |
| 31 | result.Transitions.TransitionId.Validators.Id | String | 1..1 | Идентификатор проверки |
| 32 | result.Transitions.TransitionId.Validators.Name | String | 1..1 | Описание проверки |
| 33 | result.Transitions.TransitionId.Callbacks | Object | 0..\* | Уведомления внешних систем |
| 34 | result.Transitions.TransitionId.Callbacks.Id | string | 1..1 | Идентификатор уведомления |
| 35 | result.Transitions.TransitionId.Callbacks.Name | string | 1..1 | Описание уведомления, которое уходит при успешном выполнении операции (перехода) |
| 36 | result.IsDisabled | boolean | 0..1 | Свойство блокировки маршрута. Если маршрут заблокирован, то по нему будет невозможно создать новую заявку. |

Пример:

|  |
| --- |
| { "Result": { "Id": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "Name": "string", "Description": "string", "IsConstruction": true, "AreaId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "AreaName": "string", "MetadataId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "Stages": { "StageId": { "Id": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "Name": "string", "Description": "string", "Validators": [ { "Id": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "Name": "string" } ], "IsDisabled": true, "BusinessStatus": { "System": "string", "Code": "string" } } }, "Transitions": { "TransitionId": { "Id": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "Name": "string", "FromStageId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "ToStageId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "SchemaId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "Validators": [ { "Id": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "Name": "string" } ], "Callbacks": [ { "Id": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "Name": "string" } ] } }, "IsDisabled": true, "Metadata": {} }, "Success": true, "ErrorCode": 0, "Message": "string", "StackTrace": "string"} |

# **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 | initialTransitionId | string | 1..1 | Идентификатор перехода для создания заявки |
| 4 | processContext | object | 1..1 | Набор данных |
| 5 | roleContext | object | 1..1 | Значение ролевого контекста пользователя |

Пример запроса по стандарту FHIR. Данный метод удобно использовать для получения валидного ответа согласно стандарту FHIR в случае, если маршрут поддерживает данный стандарт хранения и передачи данных. Поддержка данного стандарта указывается в метаданных маршрута ("FHIRSupported": true):

|  |
| --- |
| POST http://[hostname]/api/Fhir/StartNewProcess authorization: N3[пробел][GUID передающей системы] content-type: application/json{ "resourceType": "Parameters", "parameter": [ { "name": "workflowId", "valueString": "5c2c06cf-985b-4905-816e-bc0dec57dd9c" }, { "name": "name", "valueUrl": "Process-Fhir" }, { "name": "initialTransitionId", "valueUrl": "c0ea0245-6e72-4dba-b37c-21e4bdaffdad" }, { "name": "processContext", "resource": { "type": "collection", "entry": [ { "id": "4", "resource": { "resourceType": "Patient", "identifier": [ { "id": "1", "system": "urn:oid:1.2.643.5.1.13.2.1.1.736.1", "value": "1234123123" } ], "gender": "male", "birthDate": "1980-02-07" } }, { "id": "6", "resource": { "resourceType": "ServiceRequest", "contained": [ { "id": "1", "resourceType": "Parameters", "parameter": [ { "id": "1", "name": "Urgency", "valueCoding": { "system": "urn:oid:1.2.643.2.69.1.1.1.148.1", "code": "3" } }, { "id": "2", "name": "file", "valueAttachment": { "url": "/file/39856" } }, { "id": "3", "name": "anamnesis", "valueString": "На данном этапе наркозависимому требуется прохождение полноценного обследования для выявления всех нарушений, которые были развиты в процессе употребления запрещенных препаратов. Также данная стадия включает разработку индивидуального курса лечения. Если случай достаточно стандартный, то здесь потребуется не более недели, а вот в сложных вариантах заболевания корректировка терапии проводится постоянно на основании полученных результатов." }, { "id": "4", "name": "REGIS", "part": [ { "id": "1", "name": "IEMK", "valueCoding": { "system": "urn:oid:1.2.643.2.69.1.2.110", "code": "69161b26-7aca-4562-8878-42958e2733ce" } } ] } ] } ], "requester": { "reference": "Organization/a7ad714e-7c68-4950-ac7d-408bb68e23e9" }, "performer": [ { "id": "1", "reference": "Organization/20dfadd0-c709-43b0-a130-5a16301b0217" } ], "status": "active", "intent": "order", "category": [ { "id": "1", "coding": [ { "id": "1", "system": "urn:oid:1.2.643.2.69.1.1.1.56", "code": "108" } ] } ], "code": { "id": "1", "coding": [ { "id": "1", "system": "urn:oid:1.2.643.2.69.1.1.1.20", "code": "1" } ] }, "reasonCode": [ { "id": "1", "coding": [ { "id": "1", "system": "urn:oid:1.2.643.2.69.1.1.1.128", "code": "7" } ] } ], "occurrenceDateTime": "2020-02-07T13:28:17" } }, { "id": "7", "resource": { "resourceType": "Condition", "subject": { "reference": "Patient/f0a819e4-8d76-4aa3-b4ff-93d43c18346c" }, "code": { "coding": [ { "id": "1", "system": "urn:oid:1.2.643.2.69.1.1.1.2", "code": "T66" } ] } } } ], "resourceType": "Bundle" } }, { "name": "roleContext", "valueString": "[{\"orgId\":\"7\",\"roleId\":[\"7\",\"8\"]}]" } ]} |

Пример запроса по формату приложения:

|  |
| --- |
| POST http://[hostname]/api/Commands/StartNewProcess authorization: N3[пробел][GUID передающей системы] content-type: application/json{ "workflowId": "5c2c06cf-985b-4905-816e-bc0dec57dd9c", "name": "Новая заявка", "initialTransitionId": "c0ea0245-6e72-4dba-b37c-21e4bdaffdad", "processContext": { "type": "collection", "entry": [ { "id": "4", "resource": { "resourceType": "Patient", "identifier": [ { "id": "1", "system": "urn:oid:1.2.643.5.1.13.2.1.1.736.1", "value": "1234123123" } ], "gender": "male", "birthDate": "1980-02-07" } }, { "id": "6", "resource": { "resourceType": "ServiceRequest", "contained": [ { "id": "1", "resourceType": "Parameters", "parameter": [ { "id": "1", "name": "Urgency", "valueCoding": { "system": "urn:oid:1.2.643.2.69.1.1.1.148.1", "code": "3" } }, { "id": "2", "name": "file", "valueAttachment": { "url": "/file/39856" } }, { "id": "3", "name": "anamnesis", "valueString": "На данном этапе наркозависимому требуется прохождение полноценного обследования для выявления всех нарушений, которые были развиты в процессе употребления запрещенных препаратов. Также данная стадия включает разработку индивидуального курса лечения. Если случай достаточно стандартный, то здесь потребуется не более недели, а вот в сложных вариантах заболевания корректировка терапии проводится постоянно на основании полученных результатов." }, { "id": "4", "name": "REGIS", "part": [ { "id": "1", "name": "IEMK", "valueCoding": { "system": "urn:oid:1.2.643.2.69.1.2.110", "code": "69161b26-7aca-4562-8878-42958e2733ce" } } ] } ] } ], "requester": { "reference": "Organization/a7ad714e-7c68-4950-ac7d-408bb68e23e9" }, "performer": [ { "id": "1", "reference": "Organization/20dfadd0-c709-43b0-a130-5a16301b0217" } ], "status": "active", "intent": "order", "category": [ { "id": "1", "coding": [ { "id": "1", "system": "urn:oid:1.2.643.2.69.1.1.1.56", "code": "108" } ] } ], "code": { "id": "1", "coding": [ { "id": "1", "system": "urn:oid:1.2.643.2.69.1.1.1.20", "code": "1" } ] }, "reasonCode": [ { "id": "1", "coding": [ { "id": "1", "system": "urn:oid:1.2.643.2.69.1.1.1.128", "code": "7" } ] } ], "occurrenceDateTime": "2020-02-07T13:28:17" } }, { "id": "7", "resource": { "resourceType": "Condition", "subject": { "reference": "Patient/f0a819e4-8d76-4aa3-b4ff-93d43c18346c" }, "code": { "coding": [ { "id": "1", "system": "urn:oid:1.2.643.2.69.1.1.1.2", "code": "T66" } ] } } } ], "resourceType": "Bundle" }, "roleContext": []} |

**Ответ на запрос по формату FHIR**

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 результата 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", "id": "324f7d33-079f-4aca-819c-a106c975db56", "identifier": [ { "value": "TMC092025YW9B" } ], "basedOn": [ { "identifier": { "value": "5c2c06cf-985b-4905-816e-bc0dec57dd9c" } } ], "status": "in-progress", "businessStatus": { "coding": [ { "code": "04725db8-9147-4b5a-99d2-4a45440895bb" } ] } } } ]} |

**Ответ на запрос по формату приложения**

Описание структуры:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | ErrorCode | string | 1..1 | Код результата операции |
| 2 | Message | string | 1..1 | Описание результата операции |
| 3 | workflowId | string | 1..1 | Идентификатор маршрута |
| 4 | processId | string | 1..1 | Идентификатор заявки |
| 5 | stageId | string | 1..1 | Идентификатор текущего статуса заявки |
| 6 | humanFriendlyId | string | 1..1 | Человекоориентированный идентификатор заявки |

Пример:

|  |
| --- |
| { "workflowId": "5c2c06cf-985b-4905-816e-bc0dec57dd9c", "processId": "fe5e0c67-dbda-4129-8522-40db73e1818e", "stageId": "04725db8-9147-4b5a-99d2-4a45440895bb", "currentTransition": null, "humanFriendlyId": "TMC09202KM6F8", "validationResults": null, "success": true, "errorCode": 0, "message": null, "stackTrace": null} |

# **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 | object | 1..1 | Значение ролевого контекста пользователя |
| 2 | workflowFilter | object | 0...1 | Параметры фильтра по схеме описания метаданных маршрута |
| 3 | processFilter | object | 0...1 | Параметры фильтра по схеме описания метаданных заявки |
| 4 | StageFilter | Array | 0...1 | Фильтр по идентификаторам текущих статусов заявок |
| 5 | orderingField | String | 0...1 | Поле сортировки |
| 6 | descendingOrder | boolean | 0...1 | Направление сортировки |
| 7 | Skip | Number | 0..1 | Количество пропущенных элементов в выборке |
| 8 | Take | Number | 0..1 | Количество элементов в списке |

**Фильтр**

Объект для фильтрации по маршруту. Все поля являются опциональными (не обязательными).

Пример конструкции фильтра:

    "RoleContext": [

        {

            "Role": "DOCTOR",

            "Organization": "b19a2a66-c72c-f405-2c17-bfa785d29ffa"

        }

    ],

    "WorkflowFilter": {

        "name": "",

        "description": "",

        "metadata": {}

    },

    "ProcessFilter": {

        "name": "",

        "humanFriendlyId": "",

        "created": "",

        "updated": "",

        "workflow": "",

        "metadata": {}

    },

    "orderingField": "updated",

    "descendingOrder": true,

    "StageFilter": [],

    "Skip": 0,

    "Take": 20

RoleContext – фильтрует заявки по правилам доступа. В результате отображаются все заявки, где нет ограничений по ролевому доступу (значение RoleContext игнорируется) и заявки, где проверки данных ролевого контекста в текущем статусе позволяют пользователю выполнить операцию (значение RoleContext проверяются).

WorkflowFilter – фильтр заявок по атрибутам и метаданным маршрутов (workflows), по которым они движутся. Атрибуты и метаданные маршрута присваиваются ему при создании или редактировании администратором системы по соответствующей схеме (см. метод получения схем данных).

* Name - название маршрута (Строка, точное совпадение)
* Metadata - фильтр по объекту метаданных маршрута (Структура, правила сопоставления Json-объектов см. ниже)

ProcessFilter – фильтр заявок непосредственно по атрибутам и метаданным самой заявки. Атрибуты заявки присваиваются ей при создании и обновлении. Метаданные заявки генерируются при создании и каждом обновлении по соответствующей схеме (см. метод получения схем данных).

В приведенном ниже примере описания структуры метаданных заявки описывается набор и структура формируемых параметров заявки и пути их нахождения в её контексте (наборе структурированных данных агрегированных по ходу заявки по маршруту).

{

 "type": "object",

 "$schema": "http://json-schema.org/draft-04/schema#",

 "properties": {

 "Patient": {

 "selector": "$.entry[?(@.resource.resourceType=='Condition')].resource.subject.reference"

 },

 "Diagnosis": {

 "selector": "$.entry[?(@.resource.resourceType=='Condition')].resource.code.coding[?(@.system=='urn:oid:1.2.643.2.69.1.1.1.2')].code"

 },

 "Performer": {

 "selector": "$.entry[?(@.resource.resourceType=='ServiceRequest')].resource.performer[0].reference"

 },

 "Requester": {

 "selector": "$.entry[?(@.resource.resourceType=='ServiceRequest')].resource.requester.reference"

 },

 "occurrenceDateTime": {

 "selector": "$.entry[?(@.resource.resourceType=='ServiceRequest')].resource.occurrenceDateTime"

 }

 }

}

Пример структуры метаданных конкретной заявки по схеме приведенной выше:

{

 "Patient": "Patient/f0a819e4-8d76-4aa3-b4ff-93d43c18346c",

 "Diagnosis": "K00.1",

 "Performer": "Organization/dd4a7bf9-f853-46e3-af27-11a52ca1178b",

 "Requester": "Organization/dfe3eec2-8a79-4921-9b58-0ce03a5e6c10",

 "occurrenceDateTime": "2020-09-14T16:25:06"

}

* Name - имя процесса или значение человек ориентированного идентификатора(поиск по неполному совпадению)
* Workflow - название маршрута или его идентификатор (точное совпадение)
* Metadata - фильтр по объекту метаданных процесса (правила сопоставления Json-объектов см ниже)
* Created - дата (или период) создания процесса (описание передачи дат см. ниже)
* Updated - дата (или период) обновления процесса (описание передачи дат см. ниже)

**Правила сопоставления Json-объектов в качестве фильтров метаданных маршрута или заявки:**

В качестве фильтра принимается Json-объект, в котором путь к искомому значению будет таким же как в объекте, где производится поиск. Например, в метаданных процесса есть следующие данные:

 {

  "patientName": "Alexander",

  "diagnosis": {

    "name": "some name",

    "code": 315

  },

  "participant": {

    "name": "Ivan",

    "cpecialityCodes": [

      103,

      203,

      304

    ]

  }

}

Для доступа к полю "code" в объекте "diagnosis" нужно повторить путь к полю:

 {

  "diagnosis": {

    "code": 315

  }

}

Все выбранные данные в объекте в любом случае будут приведены к строке, и выбор сопоставление производится с помощью SQL-конструкции LIKE (поиск вхождения подстроки). Таким образом, следующее выражение будет эквивалентно предыдущему:

 {

  "diagnosis": {

    "code": "315"

  }

}

Для выбора нескольких значений, одного и того же поля можно использовать перечисление значений в массиве, однако в таком случае будет использовано полное совпадение:

{

  "diagnosis": {

    "code": [

      314,

      315,

      317

    ]

  }

}

В случае, если в объекте фильтра будет несколько полей — итоговое выражение будет собираться с учётом того, что все выражения должны быть истиной.

 {

  "patientName": [

    "Alex",

    "Sam"

  ],

  "diagnosis": {

    "name": "some name",

    "code": [

      314,

      315,

      317

    ]

  }

}

**Передача дат фильтрации:**

Для передачи периода дат ожидается строка, представляющая собой массив дат сериализованный в JSON: "[\"2020-08-11T15:30:47.972305+03:00\", \"2020-08-11T15:32:47.972305+03:00\"]". В случае если в массиве одна дата, будут выбраны все данные, в которых искомая дата будет больше чем, переданная в массиве дата. В случае если передано больше 2х дат — все кроме первых двух будут проигнорированы.

**Параметры сортировки:**

Для передачи сортировки необходимо передать имя из ответа в поле запроса **orderingField**. Поля доступные для передачи «Created», «Updated».

Для сортировки по убыванию в поле запроса **descendingOrder** нужно передать **true**.

Пример:

|  |
| --- |
| POST http://[hostname]/api/Queries/GetTransitionAvailableProcesses authorization: N3[пробел][GUID передающей системы] content-type: application/json{ "RoleContext": [ { "Role": "DOCTOR", "Organization": "Organization/dd4a7bf9-f853-46e3-af27-11a52ca1178b" } ], "WorkflowFilter": {}, "ProcessFilter": {}, "StageFilter": [], "Skip": 0, "Take": 100} |

**Ответ**

Описание структуры ответа:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | ErrorCode | string | 1..1 | Код результата операции |
| 2 | Message | string | 1..1 | Описание результата операции |
| 3 | Result.TransitionIds | array | 1..1 | Идентификаторы переходов |
| 4 | result.ProcessId | string | 1..1 | Системный идентификатор заявки |
| 5 | result.ProcessHumanFriendlyId | string | 1…\* | Человекоориентированный идентификатор заявки |
| 6 | result.Metadata | object | 1..1 | Метаданные маршрута |
| 7 | result.CurrentStage | string | 1..1 | Текущий статус заявки |
| 8 | result.WorkflowId | string | 1..1 | Идентификатор маршрута |
| 9 | result.WorkflowName | string | 1..1 | Название маршрута |
| 10 | result.ProcessName | string | 1..1 | Название заявки |
| 11 | result.BusinessStatus | object | 0..1 | Статус заявки по справочнику НСИ |
| 12 | result.BusinessStatus. System | string | 0..1 | Ссылка на справочник НСИ |
| 13 | result.BusinessStatus. Code | string | 0..1 | Код справочника НСИ |

Пример:

|  |
| --- |
| { "Result": [ { "TransitionIds": [ "3fa85f64-5717-4562-b3fc-2c963f66afa6" ], "ProcessId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "ProcessHumanFriendlyId": "string", "Metadata": {}, "CurrentStage": "string", "WorkflowId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "WorkflowName": "string", "ProcessName": "string", "BusinessStatus": { "System": "string", "Code": "string" } } ], "Success": true, "ErrorCode": 0, "Message": "string", "StackTrace": "string"} |

# **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 | object | 1..1 | Значение ролевого контекста пользователя |
| 2 | WorkflowFilter | object | 0...1 | Параметры фильтра по схеме описания метаданных маршрута |
| 3 | ProcessFilter | object | 0...1 | Параметры фильтра по схеме описания метаданных заявки |
| 4 | StageFilter | Array | 0...1 | Фильтр по идентификаторам текущих статусов заявок |
| 5 | orderingField | string | 0...1 | Поле сортировки |
| 6 | descendingOrder | boolean | 0...1 | Направление сортировки |
| 7 | Skip | number | 0..1 | Количество пропущенных элементов в выборке |
| 8 | Take | number | 0..1 | Количество элементов в списке |

**Фильтр**

Объект для фильтрации по маршруту. Все поля являются опциональными (не обязательными)

Пример конструкции фильтра:

    "RoleContext": [

        {

            "Role": "DOCTOR",

            "Organization": "b19a2a66-c72c-f405-2c17-bfa785d29ffa"

        }

    ],

    "WorkflowFilter": {

        "name": "",

        "description": "",

        "metadata": {}

    },

    "ProcessFilter": {

        "name": "",

        "humanFriendlyId": "",

        "created": "",

        "updated": "",

        "workflow": "",

        "metadata": {}

    },

    "orderingField": "updated",

    "descendingOrder": true,

    "StageFilter": [],

    "Skip": 0,

    "Take": 20

RoleContext – фильтрует заявки по правилам доступа. В результате отображаются все заявки, где нет ограничений по ролевому доступу (значение RoleContext игнорируется) и заявки, где проверки данных ролевого контекста в текущем статусе позволяют пользователю выполнить операцию (значение RoleContext проверяются).

WorkflowFilter – фильтр заявок по атрибутам и метаданным маршрутов (workflows), по которым они движутся. Атрибуты и метаданные маршрута присваиваются ему при создании или редактировании администратором системы по соответствующей схеме (см. метод получения схем данных).

* Name - название маршрута (Строка, точное совпадение)
* Metadata - фильтр по объекту метаданных маршрута (Структура, правила сопоставления Json-объектов см. ниже)

ProcessFilter – фильтр заявок непосредственно по атрибутам и метаданным самой заявки. Атрибуты заявки присваиваются ей при создании и обновлении. Метаданные заявки генерируются при создании и каждом обновлении по соответствующей схеме (см. метод получения схем данных).

В приведенном ниже примере описания структуры метаданных заявки описывается набор и структура формируемых параметров заявки и пути их нахождения в её контексте (наборе структурированных данных агрегированных по ходу заявки по маршруту).

{

 "type": "object",

 "$schema": "http://json-schema.org/draft-04/schema#",

 "properties": {

 "Patient": {

 "selector": "$.entry[?(@.resource.resourceType=='Condition')].resource.subject.reference"

 },

 "Diagnosis": {

 "selector": "$.entry[?(@.resource.resourceType=='Condition')].resource.code.coding[?(@.system=='urn:oid:1.2.643.2.69.1.1.1.2')].code"

 },

 "Performer": {

 "selector": "$.entry[?(@.resource.resourceType=='ServiceRequest')].resource.performer[0].reference"

 },

 "Requester": {

 "selector": "$.entry[?(@.resource.resourceType=='ServiceRequest')].resource.requester.reference"

 },

 "occurrenceDateTime": {

 "selector": "$.entry[?(@.resource.resourceType=='ServiceRequest')].resource.occurrenceDateTime"

 }

 }

}

Пример структуры метаданных конкретной заявки по схеме приведенной выше:

{

 "Patient": "Patient/f0a819e4-8d76-4aa3-b4ff-93d43c18346c",

 "Diagnosis": "K00.1",

 "Performer": "Organization/dd4a7bf9-f853-46e3-af27-11a52ca1178b",

 "Requester": "Organization/dfe3eec2-8a79-4921-9b58-0ce03a5e6c10",

 "occurrenceDateTime": "2020-09-14T16:25:06"

}

* Name - имя процесса или значение человек ориентированного идентификатора(поиск по неполному совпадению)
* Workflow - название маршрута или его идентификатор (точное совпадение)
* Metadata - фильтр по объекту метаданных процесса (правила сопоставления Json-объектов см ниже)
* Created - дата (или период) создания процесса (описание передачи дат см. ниже)
* Updated - дата (или период) обновления процесса (описание передачи дат см. ниже)

**Правила сопоставления Json-объектов в качестве фильтров метаданных маршрута или заявки:**

В качестве фильтра принимается Json-объект, в котором путь к искомому значению будет таким же, как в объекте, где производится поиск. Например, в метаданных процесса есть следующие данные:

{

  "patientName": "Alexander",

  "diagnosis": {

    "name": "some name",

    "code": 315

  },

  "participant": {

    "name": "Ivan",

    "cpecialityCodes": [

      103,

      203,

      304

    ]

  }

}

Для доступа к полю "code" в объекте "diagnosis" нужно повторить путь к полю:

 {

  "diagnosis": {

    "code": 315

  }

}

Все выбранные данные в объекте в любом случае будут приведены к строке, и выбор сопоставление производится с помощью SQL-конструкции LIKE (поиск вхождения подстроки). Таким образом, следующее выражение будет эквивалентно предыдущему:

 {

  "diagnosis": {

    "code": "315"

  }

}

Для выбора нескольких значений, одного и того же поля можно использовать перечисление значений в массиве, однако в таком случае будет использовано полное совпадение:

 {

  "diagnosis": {

    "code": [

      314,

      315,

      317

    ]

  }

}

В случае, если в объекте фильтра будет несколько полей — итоговое выражение будет собираться с учётом того, что все выражения должны быть истиной.

  {

  "patientName": [

    "Alex",

    "Sam"

  ],

  "diagnosis": {

    "name": "some name",

    "code": [

      314,

      315,

      317

    ]

  }

}

**Передача дат фильтрации:**

Для передачи периода дат ожидается строка, представляющая собой массив дат сериализованный в JSON: "[\"2020-08-11T15:30:47.972305+03:00\", \"2020-08-11T15:32:47.972305+03:00\"]". В случае если в массиве одна дата, будут выбраны все данные, в которых искомая дата будет больше чем, переданная в массиве дата. В случае если передано больше 2х дат — все кроме первых двух будут проигнорированы.

**Параметры сортировки:**

Для передачи сортировки необходимо передать имя из ответа в поле запроса **orderingField**. Поля доступные для передачи «Created», «Updated».

Для сортировки по убыванию в поле запроса **descendingOrder** нужно передать **true**.

Пример:

|  |
| --- |
| POST http://[hostname]/api/Queries/GetReadAvailableProcesses authorization: N3[пробел][GUID передающей системы] content-type: application/json{ "RoleContext": [ { "Role": "SENIOR\_DISPATCHER", "Organization": "b19a2a66-c72c-f405-2c17-bfa785d29ffa" } ], "WorkflowFilter": {}, "ProcessFilter": {}, "StageFilter": [“m19a2a66-t72c-v423-2c17-mfa785d29ff3”], "Skip": 0, "Take": 20}  |

**Ответ**

Описание структуры ответа:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | ErrorCode | string | 1..1 | Код результата операции |
| 2 | Message | string | 1..1 | Описание результата операции |
| 3 | result.ProcessId | string | 1..1 | Системный идентификатор заявки |
| 4 | result.ProcessHumanFriendlyId | string | 1…\* | Человекоориентированный идентификатор заявки |
| 5 | result.Metadata | object | 1..1 | Метаданные маршрута |
| 6 | result.CurrentStage | string | 1..1 | Текущий статус заявки |
| 7 | result.WorkflowId | string | 1..1 | Идентификатор маршрута |
| 8 | result WorkflowName | string | 1..1 | Название маршрута |
| 9 | result. ProcessName | string | 1..1 | Название заявки |
| 10 | result.BusinessStatus | object | 0..1 | Статус заявки по справочнику НСИ |
| 11 | result.BusinessStatus. System | string | 0..1 | Ссылка на справочник НСИ |
| 12 | result.BusinessStatus. Code | string | 0..1 | Код справочника НСИ |

Пример:

|  |
| --- |
| { "Result": [ { "ProcessId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "ProcessHumanFriendlyId": "string", "Metadata": {}, "CurrentStage": "string", "WorkflowId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "WorkflowName": "string", "ProcessName": "string", "BusinessStatus": { "System": "string", "Code": "string" } } ], "Success": true, "ErrorCode": 0, "Message": "string", "StackTrace": "string"} |

# **Получение заявки по идентификатору**

**Описание параметров запроса**

Метод возвращает данные о заявке по её идентификатору.

**Запрос**

Пример:

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

**Ответ**

Описание структуры ответа:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | ErrorCode | string | 1..1 | Код результата операции |
| 2 | Message | string | 1..1 | Описание результата операции |
| 3 | result.id | string | 1..1 | Системный идентификатор заявки |
| 4 | humanFriendlyId | string | 1…\* | Человекоориентированный идентификатор заявки |
| 5 | result.metadataId | string | 1..1 | Метаданные маршрута |
| 6 | result.currentStageId | string | 1..1 | Текущий статус заявки |
| 7 | result. workflowId | string | 1..1 | Идентификатор маршрута |
| 8 | result.name | string | 1..1 | Название заявки |
| 9 | result.created | string | 0..1 | Время создания |
| 10 | result.updated | string | 0..1 | Время последнего изменения |
| 11 | result.BusinessStatus | object | 0..1 | Статус заявки по справочнику НСИ |
| 12 | result.BusinessStatus. System | string | 0..1 | Ссылка на справочник НСИ |
| 13 | result.BusinessStatus. Code | string | 0..1 | Код справочника НСИ |

Пример:

|  |
| --- |
| { "result": { "id": "bdcae5e4-e945-4d17-902a-d846864a6adc", "name": "Новая заявка", "workflowId": "5c2c06cf-985b-4905-816e-bc0dec57dd9c", "metadataId": "ff53e341-641f-43c6-86bd-43b47e863d94", "currentStageId": "04725db8-9147-4b5a-99d2-4a45440895bb", "created": "2020-09-01T16:15:58.41756+03:00", "updated": "2020-09-01T16:15:58.41756+03:00", "businessStatus": { "system": "urn:oid:1.2.643.2.69.1.1.1.148.2", "code": "1" }, "humanFriendlyId": "TMC0920VWV523" }, "success": true, "errorCode": 0, "message": null, "stackTrace": null} |

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

**Описание параметров запроса**

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

Описание структуры запроса:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | RoleContext | object | 1..1 | Структуры ролевого контекста согласно схемам предметной области |
| 2 | ProcessId | string | 1..1 | Идентификатор процесса () |

**Запрос**

Пример запроса по стандарту FHIR. Данный метод удобно использовать для получения валидного ответа согласно стандарту FHIR в случае, если маршрут поддерживает данный стандарт хранения и передачи данных. Поддержка данного стандарта указывается в метаданных маршрута ("FHIRSupported": true):

|  |
| --- |
| POST http://[hostname]/api/Fhir/GetProcessContext?processId=[id заявки] authorization: N3[пробел][GUID передающей системы] content-type: application/json{ "resourceType": "Parameters", "parameter": [ { "name": "processId", "valueString": "b894cd31-76be-4269-9e0b-0f2eab7d3e6e" }, { "name": "roleContext", "valueString": "{\"orgId\":\"7\",\"roleId\":[\"7\",\"8\"]}" } ]} |

Пример запроса по формату приложения:

|  |
| --- |
| POST http://[hostname]/api/Queries/GetProcessContext?processId=[id заявки] authorization: N3[пробел][GUID передающей системы] content-type: application/json{ "RoleContext": [ { "Role": "PARAMEDIC", "Organization": "09f4c585-13de-4f6a-b16a-fd3f8827d8b1" } ], "ProcessId": "25b9cedf-f3a7-40be-9864-c54080a5c8f5"} |

**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 результата операции |
| 3 | expression | string | 0..\* | Объект контекста, структурированный по схеме предметной области |

Пример ответа по формату Fhir:

|  |
| --- |
| { "resourceType": "Bundle", "type": "transaction-response", "entry": [ { "resource": { "resourceType": "OperationOutcome", "issue": [ { "severity": "information", "code": "informational", "diagnostics": "<значение errorCode>" } ] } }, { "resource": {<содержание заявки>} } ]} |

Пример ответа по формату приложения:

|  |
| --- |
| { "result": {<содержание заявки>}, "success": true, "errorCode": 0, "message": null, "stackTrace": null} |

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

**Описание параметров запроса**

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

**Запрос**

Описание:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | roleContext | object | 1..1 | Значение ролевого контекста пользователя |
| 2 | ProcessId | string | 1..1 | Системный идентификатор заявки |

Пример:

|  |
| --- |
| POST http://[hostname]/api/Queries/GetProcessWithAvailableTransitionsauthorization: N3[пробел][GUID передающей системы] content-type: application/json{ "ProcessId": "25b9cedf-f3a7-40be-9864-c54080a5c8f5", "RoleContext": [ { "system": "N3 1a55e61b-dd7a-4acf-a94a-37d21d761af5" } ]} |

**Ответ**

Описание структуры OperationOutcome:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | ErrorCode | string | 1..1 | Код результата операции |
| 2 | Message | string | 1..1 | Описание результата операции |
| 3 | Result. availableTransitions | array | 1..\* | Доступные переходы для данной заявки |
| 4 | Result. availableTransitions. id | string | 1..1 | Идентификатор перехода |
| 5 | Result. availableTransitions. Name | string | 1..1 | Название перехода |
| 6 | Result. availableTransitions. fromStageId | string | 1..1 | Стартовый статус |
| 7 | Result. availableTransitions. toStageId | string | 1..1 | Итоговый статус |
| 8 | Result. availableTransitions. schemaId | string | 1..1 | Идентификатор схемы данных необходимых для осуществления перехода |
| 9 | Result. availableTransitions. validatorIds | string | 1..1 | Массив проверок выполняемых при осуществлении перехода |
| 10 | Result. availableTransitions. callbackIds | Array | 1..1 | Массив внешних вызовов осуществляемых при переходе |
| 11 | Result. availableTransitions. roleSchemaIds | Array | 1..1 | Массив схем данных описывающих ролевой контекст пользователей, работающих с данным переходом |
| 12 | Result.id | string | 1..1 | Идентификатор заявки |
| 13 | Result.name | string | 1..1 | Название заявки |
| 14 | Result. workflowId | string | 1..1 | Идентификатор маршрута заявки |
| 15 | Result. metadataId | string | 1..1 | Идентификатор метаданных заявки |
| 16 | Result. currentStageId | string | 1..1 | Идентификатор текущего статуса заявки |
| 17 | Result. created | string | 1..1 | Время создания заявки |
| 18 | Result. updated | string | 1..1 | Время последнего обновления заявки  |
| 19 | Result. businessStatus | object | 1..1 | Статус заявки по справочнику НСИ |
| 20 | Result. businessStatus. system | string | 1..1 | Ссылка на справочник НСИ |
| 21 | Result. businessStatus.code | string | 1..1 | Код справочника НСИ |
| 22 | Result. humanFriendlyId | string | 1..1 | Человекоориентированный идентификатор заявки |

Пример:

|  |
| --- |
| { "Result": { "AvailableTransitions": [ { "Id": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "Name": "string", "FromStageId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "ToStageId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "SchemaId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "ValidatorIds": [ "3fa85f64-5717-4562-b3fc-2c963f66afa6" ], "CallbackIds": [ "3fa85f64-5717-4562-b3fc-2c963f66afa6" ], "RoleSchemaIds": [ "3fa85f64-5717-4562-b3fc-2c963f66afa6" ] } ], "Id": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "Name": "string", "WorkflowId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "MetadataId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "CurrentStageId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "Created": "2020-09-02T15:15:30.676Z", "Updated": "2020-09-02T15:15:30.676Z", "BusinessStatus": { "System": "string", "Code": "string" }, "HumanFriendlyId": "string" }, "Success": true, "ErrorCode": 0, "Message": "string", "StackTrace": "string"} |

# **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 ответа:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | ErrorCode | string | 1..1 | Код результата операции |
| 2 | Message | string | 1..1 | Описание результата операции |
| 4 | Result.id | string | 1..1 | Идентификатор перехода |
| 5 | Result.Name | string | 1..1 | Название перехода |
| 6 | Result.fromStageId | string | 1..1 | Стартовый статус |
| 7 | Result.toStageId | string | 1..1 | Итоговый статус |
| 8 | Result.schemaId | string | 1..1 | Идентификатор схемы данных необходимых для осуществления перехода |

Пример:

|  |
| --- |
| { "result": { "id": "b779687c-2f32-4d8d-879e-478e81af6607", "name": "Создать заявку", "fromStageId": null, "toStageId": "617690fd-de03-41d6-b2df-793f765ef537", "schemaId": "be371120-57fd-402f-9ed6-5a89422d074f", "validatorIds": [ "6790c352-5e6d-4d88-b630-73c5cc92e4a3" ], "callbackIds": [], "roleSchemaIds": [ "b0f07058-9a15-4235-bc9d-2c132d88a17c" ] }, "success": true, "errorCode": 0, "message": null, "stackTrace": null} |

# **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 ответа:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | ErrorCode | string | 1..1 | Код результата операции |
| 2 | Message | string | 1..1 | Описание результата операции |
| 3 | Result.id | string | 1..1 | Идентификатор перехода |
| 4 | Result.Name | string | 1..1 | Название перехода |
| 5 | Result.Type | number | 1..1 | Тип схемы: /// Основной тип схемы. Primary = 0, /// Схема метаданных заявки. Process = 1, /// Схема метаданных маршрута. Workflow = 2, /// Схема метаданных перехода. Transition = 3, /// Схема информации о пользователе RoleContext = 4 |
| 6 | Result.AreaId | string | 1..1 | Идентификатор предметной области, в рамках которой задана схема данных |
| 7 | Result.AreaName | string | 1..1 | Название предметной области, в рамках которой задана схема данных |
| 8 | Result.Schema | object | 1..1 | Json-схема |
| 9 | Result.IsDisabled | boolean | 1..1 | Статус блокировки |

Пример:

|  |
| --- |
| { "Result": { "Id": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "Name": "string", "Type": 0, "AreaId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "AreaName": "string", "Schema": {}, "IsDisabled": true }, "Success": true, "ErrorCode": 0, "Message": "string", "StackTrace": "string"} |

# **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 | object | 1..1 | Набор данных |
| 4 | roleContext | object | 1..1 | Значение ролевого контекста пользователя |

Пример запроса по формату FHIR:

|  |
| --- |
| POST http://[hostname]/api/Fhir/MoveToStageauthorization: N3[пробел][GUID передающей системы] content-type: application/json{ "resourceType": "Parameters", "parameter": [ { "name": "processId", "valueString": "47fe3764-957c-47ab-9fbd-76c48962f62e" }, { "name": "transitionId", "valueUrl": "1371cd97-9008-4a2d-b7b3-abc0f153cf6f" }, { "name": "processContext", "resource": {} }, { "name": "roleContext", "valueString": "[{ \"Role\": \"DOCTOR\",\"Organization\": \"Organization/20dfadd0-c709-43b0-a130-5a16301b0217\"}]" } ]} |

Пример запроса по формату приложения:

|  |
| --- |
| POST http://[hostname]/api/Commands/MoveToStageauthorization: N3[пробел][GUID передающей системы] content-type: application/json{ "processId": "8f3baaea-531a-409a-a690-7401bef83eb9", "transitionId": "1371cd97-9008-4a2d-b7b3-abc0f153cf6f", "processContext": { "type": "collection", "entry": [ { "id": "1", "resource": { "resourceType": "DiagnosticReport", "status": "final", "code": { "coding": [ { "id": "1", "system": "urn:oid:1.2.643.5.1.13.13.99.2.41", "code": "8" } ] }, "conclusion": "Some text", "presentedForm": [ { "id": "1", "url": "http://10.62.1.28:6789/integration/zak", "title": "Заключение" } ] } } ], "resourceType": "Bundle" }, "roleContext": [ { "Role": "DOCTOR", "Organization": "Organization/20dfadd0-c709-43b0-a130-5a16301b0217" } ]} |

**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 результата операции |

Описание структуры Task:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | id | 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 ответа по формату FHIR:

|  |
| --- |
| { "resourceType": "Bundle", "type": "transaction-response", "entry": [ { "resource": { "resourceType": "OperationOutcome", "issue": [ { "severity": "information", "code": "informational", "diagnostics": "0" } ] } }, { "resource": { "resourceType": "Task", "id": "47fe3764-957c-47ab-9fbd-76c48962f62e", "identifier": [ { "value": "TMC0920S4MF81" } ], "basedOn": [ { "identifier": { "value": "5c2c06cf-985b-4905-816e-bc0dec57dd9c" } } ], "status": "in-progress", "businessStatus": { "coding": [ { "code": "cf4639a2-435f-498e-b97e-87507f7b431f" } ] } } } ]} |

**Ответ на запрос по формату приложения**

Описание структуры:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | ErrorCode | string | 1..1 | Код результата операции |
| 2 | Message | string | 1..1 | Описание результата операции |
| 3 | workflowId | string | 1..1 | Идентификатор маршрута |
| 4 | processId | string | 1..1 | Идентификатор заявки |
| 5 | stageId | string | 1..1 | Идентификатор текущего статуса заявки |
| 6 | humanFriendlyId | string | 1..1 | Человекоориентированный идентификатор заявки |

Пример ответа по формату приложения:

|  |
| --- |
| { "workflowId": "5c2c06cf-985b-4905-816e-bc0dec57dd9c", "processId": "8f3baaea-531a-409a-a690-7401bef83eb9", "stageId": "cf4639a2-435f-498e-b97e-87507f7b431f", "currentTransition": null, "humanFriendlyId": "TMC0920768GGD", "validationResults": null, "success": true, "errorCode": 0, "message": null, "stackTrace": null} |

# **Получение ссылок на чаты и видеоконференцсвязь для участников заявки**

**Конфигурируемые данные**

Для организации взаимодействия клиент-системы необходимо хранение конфигурируемых данных, к которым относятся следующие данные, определяемые провайдером:

адрес портала, обеспечивающего организацию доступа к мессенджеру (baseUrl);

ключ доступа к методам API (apiKey)

идентификатор процесса в сервисе «Сервис управления Маршрутами».

**Запрос**

Описание:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| 1 | baseUrl | string | 1..1 | Адрес портала, обеспечивающего организацию доступа к мессенджеру |
| 2 | apiKey | string | 1..1 | Ключ доступа к методам |
| 3 | process\_id | string | 1..1 | Идентификатор процесса в сервисе «Сервис управления Маршрутами». Хранится в контексте заявки на портале Телемедицины |

Для получения ссылок на чат/комнату ВКС для каждого участника по заявке клиент-системой необходимо выполнить запросом с бэкэнда с помощью метода “api/flows/process\_links” (GET).

Пример:

|  |
| --- |
| [base\_url]/api/flows/process\_links?process\_id=[process\_id]&apikey=[apiKey] |

**Ответ**

Пример:

|  |
| --- |
| { "resourceType": "Bundle", "type": "collection", "entry": [ { "resource": { "resourceType": "Parameters", "parameter": [ { "name": "Endpoint",  "valueString": "flows/process\_links" }, { "name": "process\_id",  "valueString": "3962" }, { "name": "tread\_id",  "valueString": "5353" } ] } }, { "resource": { "resourceType": "Patient", "identifier": [ { "system": "urn:oid:1.2.643.2.69.1.2.111",  "value": "f0a819e4-8d76-4aa3-b4ff-93d43c18346c"  }, { "system": "urn:oid:1.2.643.2.69.1.2.116",  "value": "6564"  } ], "name": [ { "family": "Петров",  "given": [ "Николай",  "Сергеевич" ] } ], "telecom": [ { "id": "linc\_chat", "value": "https//..."  }, { "id": "linc\_room", "value": "https//..." } ] } }, { "resource": { "resourceType": "Practitioner", "identifier": [ { "system": "urn:oid:1.2.643.2.69.1.1.1.6.22",  "value": "12345678901"  }, { "system": "urn:oid:1.2.643.2.69.1.2.116",  "value": "007"  } ], "name": [ { "family": "Пушкин",  "given": [ "Александр",  "Сергеевич"  ] } ] } }, { "resource": {  "resourceType": "PractitionerRole", "practitioner": { "reference": "Practitioner/007"  }, "code": [ { "coding": [ { "system": "1.2.643.2.69.1.1.1.56",  "code": "3"  } ] } ], "telecom": [ { "id": "linc\_chat", "value": "https//..."  }, { "id": "linc\_room", "value": "https//..."  } ] } } ]} |

**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 |

# **Контрольный пример**

Адрес стенда для осуществления отладки:

<http://r78-rc.zdrav.netrika.ru/tm-core>

Тестовый маршрут:

5c2c06cf-985b-4905-816e-bc0dec57dd9c

Состояния маршрута:

|  |  |
| --- | --- |
| Название | Идентификатор |
| В работе | 04725db8-9147-4b5a-99d2-4a45440895bb |
| Отказано | 91a086e3-e46e-4a44-9d0d-6676e6884f3e |
| Заключение готово | cf4639a2-435f-498e-b97e-87507f7b431f |

**Примеры запросов**

 "workflowId": "346f9527-6a86-4c1b-8076-19cc58f463e5",

 "initialTransitionId": "270ca374-6e43-48bf-b545-4f01d45118d0",

 "processContext":{<смотри пример ниже>},

 "roleContext": []

**Возможные переходы между статусами в заявке**

|  |  |  |  |
| --- | --- | --- | --- |
| Название операции | Идентификатор операции | Стартовый статус | Итоговый статус |
| Создание заявки | c0ea0245-6e72-4dba-b37c-21e4bdaffdad |  | В работе |
| Отклонение заявки врачом | fc619e45-2a25-4650-ae30-34ef63296c44 | В работе | Отказано |
| Отправка заключения | 1371cd97-9008-4a2d-b7b3-abc0f153cf6f | В работе | Заключение готово |

**Примеры контекстов данных переходов**

|  |  |
| --- | --- |
| Название перехода | Пример processContext |
| Создание заявки | { "type": "collection", "entry": [ { "id": "4", "resource": { "resourceType": "Patient", "identifier": [ { "id": "1", "system": "urn:oid:1.2.643.5.1.13.2.1.1.736.1", "value": "1234123123" } ], "gender": "male", "birthDate": "1980-02-07" } }, { "id": "6", "resource": { "resourceType": "ServiceRequest", "contained": [ { "id": "1", "resourceType": "Parameters", "parameter": [ { "id": "1", "name": "Urgency", "valueCoding": { "system": "urn:oid:1.2.643.2.69.1.1.1.148.1", "code": "3" } }, { "id": "2", "name": "file", "valueAttachment": { "url": "/file/39856" } }, { "id": "3", "name": "anamnesis", "valueString": "На данном этапе наркозависимому требуется прохождение полноценного обследования для выявления всех нарушений, которые были развиты в процессе употребления запрещенных препаратов. Также данная стадия включает разработку индивидуального курса лечения. Если случай достаточно стандартный, то здесь потребуется не более недели, а вот в сложных вариантах заболевания корректировка терапии проводится постоянно на основании полученных результатов." }, { "id": "4", "name": "REGIS", "part": [ { "id": "1", "name": "IEMK", "valueCoding": { "system": "urn:oid:1.2.643.2.69.1.2.110", "code": "69161b26-7aca-4562-8878-42958e2733ce" } } ] } ] } ], "requester": { "reference": "Organization/a7ad714e-7c68-4950-ac7d-408bb68e23e9" }, "performer": [ { "id": "1", "reference": "Organization/20dfadd0-c709-43b0-a130-5a16301b0217" } ], "status": "active", "intent": "order", "category": [ { "id": "1", "coding": [ { "id": "1", "system": "urn:oid:1.2.643.2.69.1.1.1.56", "code": "108" } ] } ], "code": { "id": "1", "coding": [ { "id": "1", "system": "urn:oid:1.2.643.2.69.1.1.1.20", "code": "1" } ] }, "reasonCode": [ { "id": "1", "coding": [ { "id": "1", "system": "urn:oid:1.2.643.2.69.1.1.1.128", "code": "7" } ] } ], "occurrenceDateTime": "2020-02-07T13:28:17" } }, { "id": "7", "resource": { "resourceType": "Condition", "subject": { "reference": "Patient/f0a819e4-8d76-4aa3-b4ff-93d43c18346c" }, "code": { "coding": [ { "id": "1", "system": "urn:oid:1.2.643.2.69.1.1.1.2", "code": "T66" } ] } } } ], "resourceType": "Bundle"}  |
| Отклонение заявки регистратором | { "type": "collection", "entry": [ { "id": "1", "resource": { "resourceType": "DiagnosticReport", "status": "final", "code": { "coding": [ { "id": "1", "system": "urn:oid:1.2.643.5.1.13.13.99.2.41", "code": "8" } ] }, "conclusion": "Some text", "conclusionCode": [ { "id": "1", "coding": [ { "id": "1", "system": "urn:oid:1.2.643.2.69.1.1.1.132", "code": "6" } ] } ] } } ], "resourceType": "Bundle"}  |
| Отправка заключения | { "type": "collection", "entry": [ { "id": "1", "resource": { "resourceType": "DiagnosticReport", "status": "final", "code": { "coding": [ { "id": "1", "system": "urn:oid:1.2.643.5.1.13.13.99.2.41", "code": "8" } ] }, "conclusion": "Some text", "presentedForm": [ { "id": "1", "url": "http://10.62.1.28:6789/integration/zak", "title": "Заключение" } ] } } ], "resourceType": "Bundle" }  |

**Расшифровка параметров processContext запросов**

Создание заявки

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Jsonpath в структуре передаваемых данных | Обязательность | Тип данных |
| Паспорт пациента | $.entry[?(@.resource.resourceType=='Patient')].resource.identifier[?(@.system=='urn:oid:1.2.643.5.1.13.2.1.1.736.1')].value | 1...1 | строка |
| Пол пациента | $.entry[?(@.resource.resourceType=='Patient')].resource.gender | 1...1 | строка |
| Дата рождения пациента | $.entry[?(@.resource.resourceType=='Patient')].resource. birthDate | 1...1 | строка |
|  |  |  |  |
| Срочность  | $.entry[?(@.resource.resourceType=='ServiceRequest')].resource.contained[\*].parameter[?(@.name=='Urgency')].valueCoding.code | 1...1 | строка |
| Приложенный файл | $.entry[?(@.resource.resourceType=='ServiceRequest')].resource.contained[\*].parameter[?(@.name=='file')].valueAttachment.url | 0...1 | строка |
| Комментарий к заявке (описание проблемы) | $.entry[?(@.resource.resourceType=='ServiceRequest')].resource.contained[\*].parameter[?(@.name=='anamnesis')].valueString | 1...1 | строка |
| Направляющая организация | $.entry[?(@.resource.resourceType=='ServiceRequest')].resource.requester.reference | 1...1 | строка |
| Целевая организация | $.entry[?(@.resource.resourceType=='ServiceRequest')].resource.performer[\*].reference | 1...1 | строка |
| Медицинский профиль | $.entry[?(@.resource.resourceType=='ServiceRequest')].resource.category[\*].coding[?(@.system=='urn:oid:1.2.643.2.69.1.1.1.56')].code | 1...1 | строка |
| Первичность обращения пациента | $.entry[?(@.resource.resourceType=='ServiceRequest')].resource.code.coding[?(@.system=='urn:oid:1.2.643.2.69.1.1.1.20')].code | 1...1 | строка |
| Цель | $.entry[?(@.resource.resourceType=='ServiceRequest')].resource.reasonCode[\*].coding[?(@.system=='urn:oid:1.2.643.2.69.1.1.1.128')].code | 1...1 | строка |
| Желаемое время консультации | $.entry[?(@.resource.resourceType=='ServiceRequest')].resource.occurrenceDateTime | 1...1 | строка |
|  |  |  |  |
| Идентификатор пациента MPI | $.entry[?(@.resource.resourceType=='Condition')].resource.subject.reference | 1...1 | строка |
| Диагноз | $.entry[?(@.resource.resourceType=='Condition')].resource.code.coding[?(@.system=='urn:oid:1.2.643.2.69.1.1.1.2')].code | 1...1 | строка |

Отклонение заявки

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Поле в структуре | Обязательность | Тип данных |
| Комментарий | $.entry[?(@.resource.resourceType=='DiagnosticReport')].resource.conclusion | 1...1 | строка |
| Причина отклонения | $.entry[?(@.resource.resourceType=='DiagnosticReport')].resource.conclusionCode[\*].coding[?(@.system=='urn:oid:1.2.643.2.69.1.1.1.132')].code | 1...1 | строка |

Заключение

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Поле в структуре | Обязательность | Тип данных |
| Текст заключения | $.entry[?(@.resource.resourceType=='DiagnosticReport')].resource.conclusion | 1...1 | строка |
| Файлы заключения | $.entry[?(@.resource.resourceType=='DiagnosticReport')].resource.presentedForm[\*].url | 1...1 | строка |