|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
|  |  | | | |  |
|  | | | | | |
|  | **Техническое приложение к регламенту информационного взаимодействия** | | | |  |
| Описание интеграционных профилей  Сервис активы | | | | | |
|  | | | | | |
|  | | Листов 77 | |  | |
|  | | |  | | |
|  | | |  | | |
|  | | | | | |

Аннотация

Настоящий документ описывает регламент работы с сервисом интеграционной платформы N3 «Активы» (Сервис Активы).

Указанный сервис обеспечивает механизмы взаимодействия и обмена медицинскими данными между различными информационными системами, обслуживающими процесс передачи сведений о заявках на обслуживание активных вызовов при оказании медицинской помощи населению.

В состав описания включены схемы процессов, описание используемых технологий, методов, входных и выходных данных.

**История версий**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Версия | Дата | Автор | Примечание |
|  | 1.0 | 30/11/2020 | Павлова А.Н. | Первая версия регламента |
|  | 2.0 | 01/12/2020 | Павлова А.Н. | Добавлено описание параметров по fhir |
|  | 3.0 | 3/12/2020 | Павлова А.Н. | Правка текста. Поправлены примеры. |
|  | 4.0 | 9/12/2020 | Павлова А.Н. | Внесены правки в описание параметров согласно изменениям структуры данных обмена |
|  | 5.0 | 11/12/2020 | Павлова А.Н. | Изменено наименование параметра направляющей МО |
|  | 6.0 | 18/12/2020 | Павлова А.Н. | Добавлены примеры и скорректировано описание фильтров |
|  | 7.0 | 27/01/2021 | Павлова А.Н. | Добавлено описание методов работы с файлами. |
|  | 8.0 | 27/01/2021 | Павлова А.Н. | Добавлено описание принципа автоматического преобразования данных к FHIR - стандарту |
|  | 9.0 | 29/06/2021 | Павлова А.Н. | Добавлено описание метода получения схемы на переходе |
|  | 10.0 | 29/06/2021 | Павлова А.Н. | Добавлены параметры ответов методов создания и изменения заявки |

**Содержание**

[Аннотация 2](#_Toc65075594)

[История версий 3](#_Toc65075595)

[Содержание 4](#_Toc65075596)

[1. Общие положения 7](#_Toc65075597)

[2. Определения, обозначения и сокращения 8](#_Toc65075598)

[3. Описание решения 9](#_Toc65075599)

[3.1. Описание процесса обмена данными о заявках на обслуживание активного вызова 9](#_Toc65075600)

[3.2. Описание взаимодействия с сервисом 11](#_Toc65075601)

[4. Описание протокола взаимодействия 13](#_Toc65075602)

[4.1. Общая информация о сервисе 13](#_Toc65075603)

[4.2. Использование справочников 13](#_Toc65075604)

[4.3. Внутренняя логика сервиса 13](#_Toc65075605)

[**Описание данных** 13](#_Toc65075606)

[**Маршрут и его описание** 14](#_Toc65075607)

[**Переходы и ролевой доступ к просмотру** 17](#_Toc65075608)

[5. Описание методов 19](#_Toc65075609)

[5.1. Состав методов сервиса 19](#_Toc65075610)

[5.2. Структура данных заявки 19](#_Toc65075611)

[5.3. Структура метаданных заявки 20](#_Toc65075612)

[5.4. Описание параметров объектов, входящих в структуру заявки 23](#_Toc65075613)

[5.5. Создание заявки (POST StartNewProcess) 30](#_Toc65075614)

[**5.5.1.** **Описание параметров запроса** 30](#_Toc65075615)

[**5.5.2.** Пример запроса 30](#_Toc65075616)

[5.6. Редактирование/обновление заявки (POST MoveToStage) 30](#_Toc65075617)

[**5.6.1.** **Описание параметров запроса на редактирвание заявки** 31](#_Toc65075618)

[**5.6.2.** Примеры запроса 31](#_Toc65075619)

[5.7. Получение списка доступных для действия заявок (POST //GetTransitionAvailableProcesses) 31](#_Toc65075620)

[**5.7.1.** **Описание параметров запроса** 31](#_Toc65075621)

[**5.7.2.** Пример запроса 33](#_Toc65075622)

[**5.7.3.** Пример ответа 34](#_Toc65075623)

[5.8. Получение списка доступных для просмотра заявок (POST //GetReadAvailableProcesses). 36](#_Toc65075624)

[**5.8.1.** **Описание параметров запроса** 36](#_Toc65075625)

[**5.8.2.** Пример запроса 37](#_Toc65075626)

[**5.8.3.** Пример ответа 38](#_Toc65075627)

[5.9. Получение объекта контекста заявки (POST //GetProcessContext) 40](#_Toc65075628)

[**5.9.1.** **Описание параметров запроса** 40](#_Toc65075629)

[**5.9.2.** Пример запроса 40](#_Toc65075630)

[5.10. Получение маршрута обработки заявки по идентификатору (GetWorkflow) 41](#_Toc65075631)

[**5.10.1.** **Описание параметров запроса и ответа** 41](#_Toc65075632)

[**5.10.2.** Пример запроса 43](#_Toc65075633)

[5.11. Организация обмена файлами. Компонент XDS 43](#_Toc65075634)

[**5.11.1.** **Передача файла вложения заявки (POST //api/Commands/xds)** 43](#_Toc65075635)

[**5.11.2.** **Описание параметров запроса** 44](#_Toc65075636)

[**5.11.3.** Пример запроса 44](#_Toc65075637)

[**5.11.4.** **Получение файла вложения заявки (POST //xds)** 44](#_Toc65075638)

[**5.11.5.** Описание параметров запроса 44](#_Toc65075656)

[**5.11.6.** Пример запроса 45](#_Toc65075657)

[5.12. Организация поддержки FHIR-стандарта в методах сервиса 45](#_Toc65075658)

[**5.12.1.** **Общий принцип преобразования** 45](#_Toc65075659)

[**5.12.2.** **Обработка данных из processContext в FHIR** 47](#_Toc65075660)

[**5.12.3.** **Обработка данных из roleContext в FHIR** 53](#_Toc65075661)

[**5.12.4.** **Описание метода создания заявки по стандарту FHIR (POST //api/Fhir/StartNewProcess)** 54](#_Toc65075662)

[5.12.4.1. Пример формата запроса по стандарту Fhir 55](#_Toc65075663)

[**5.12.5.** **Описание метода редактирования/обновления заявки по стандарту FHIR (POST //api/Fhir/MoveToStage)** 55](#_Toc65075664)

[5.12.5.1. Пример формата запроса по стандарту Fhir 56](#_Toc65075665)

[**5.12.6.** **Описание метода получения объекта контекста заявки по стандарту FHIR (POST //api/Fhir/ProcessContext)** 56](#_Toc65075666)

[5.12.6.1. Пример формата запроса по стандарту Fhir 56](#_Toc65075667)

[**5.12.7.** Пример полного преобразования тела запроса по созданию заявки в FHIR 57](#_Toc65075668)

[6. Коды возвращаемых ошибок 78](#_Toc65075669)

# Общие положения

* 1. Настоящее описание интеграционных профилей Сервиса Активы (далее – Описание) определяет механизмы информационного взаимодействия медицинских информационных систем (далее МИС) и сервиса Активы для обмена данными заявок на обслуживание активных вызовов.
  2. Описание предназначено для организаций-разработчиков, осуществляющих сопровождение эксплуатируемых информационных систем и разработку новых систем для медицинских учреждений.
  3. Документ содержит описание методов сервиса Активы, которые должны поддерживать сторонние информационные системы для обеспечения автоматизированного информационного взаимодействии.

# Определения, обозначения и сокращения

Таблица 1 Определения и сокращения

|  |  |
| --- | --- |
| **Термин** | **Содержание** |
| ГИС РЕГИЗ | Государственная информационная система Краснодарского края «Региональный фрагмент единой государственной информационной системой в сфере здравоохранения» |
| МО | Медицинская организация |
| МИС МО | Медицинская информационная система медицинской организации |
| НСИ | Подсистема ГИС РЕГИЗ «Нормативно-справочная информация» |
| ТМ.РЕГИЗ | Подсистема «Телемедицина» государственной информационной системы «Региональный фрагмент единой государственной информационной системы в сфере здравоохранения» |
| Индекс пациента (MPI) | Подсистема ГИС РЕГИЗ «Индекс пациента» |
| СНИЛС | Страховой номер индивидуального лицевого счёта, СНИЛС — уникальный номер индивидуального лицевого счёта застрахованного лица в системе обязательного пенсионного страхования |
| СМП | Скорая медицинская помощь |
| ЧСС | Частота сердечных сокращений |
| АД | Артериальное давление |
| МКБ-10 | Международный классификатор болезней |
| ЛПУ | Лечебно-профилактическое учреждение |
| ЭЦП | Электронная цифровая подпись |

# Описание решения

## Описание процесса обмена данными о заявках на обслуживание активного вызова

Для заявок на обслуживание активных вызовов в роли направляющей МО выступает СМП, в роли целевой – МО амбулаторного профиля.

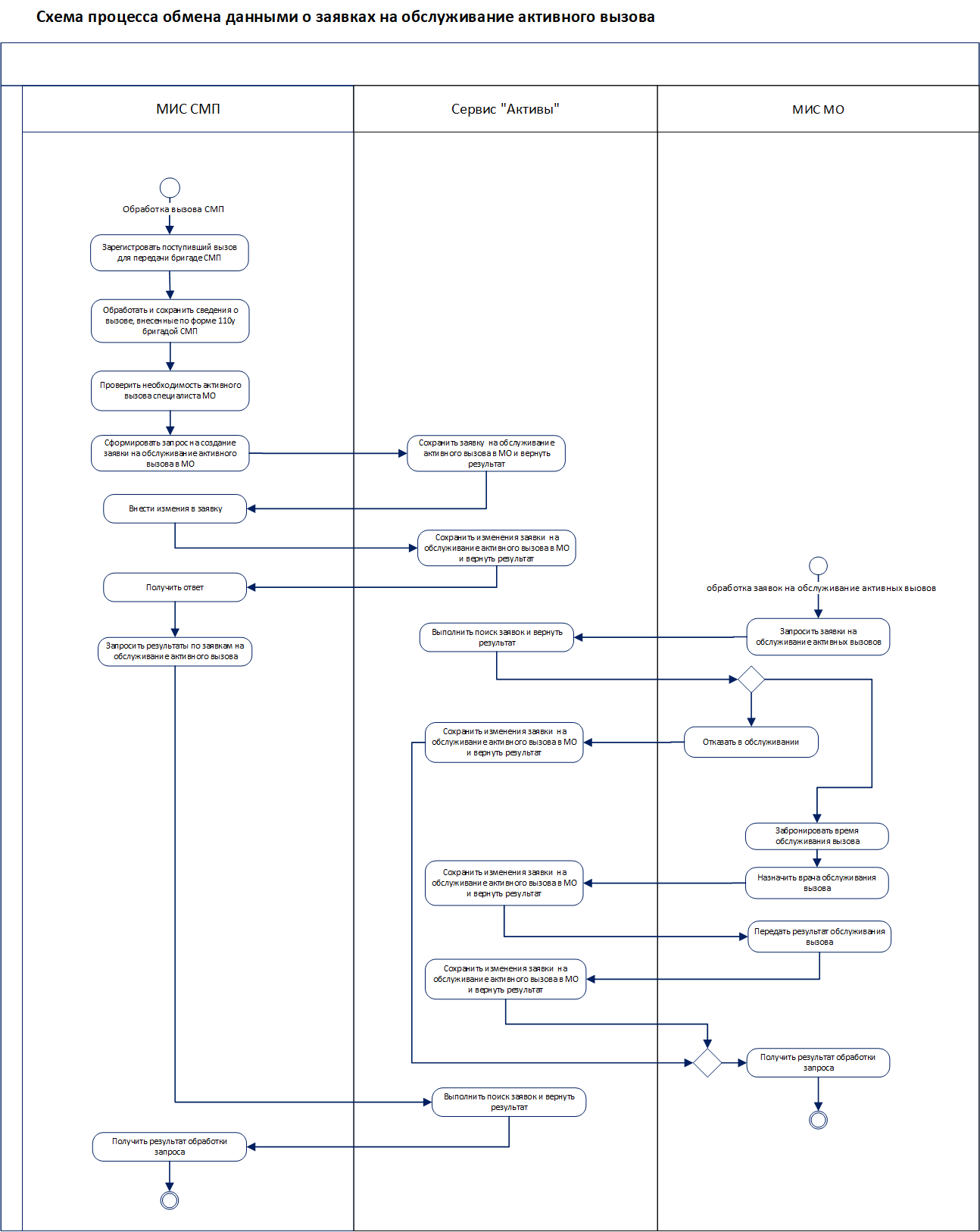
В рамках процесса обслуживания активного выхода обмен данными ведется, исходя из требований нормативного документа статистического учета «Учетная форма N 110/у «Карта вызова скорой медицинской помощи»».

Процесс обмена данными о заявках на обслуживание активного вызова подразумевает прохождение следующих этапов:

* МИС СМП регистрирует вызов для последующей передачи данных бригаде СМП.
* После обслуживания бригадой СМП информация о вызове дополняется в МИС СМП новыми данными в объеме, определенном требованиями нормативного документа.
* МИС СМП обрабатывает информацию и формирует к отправке заявки на обслуживание активных вызовов для последующей передачи в целевые МО.
* МИС МО со своей стороны запрашивает список заявок на обслуживание активных вызовов и обрабатывает результат.
* В МИС МО бронируется временной период для обслуживания вызова. МИС МО передает данные о назначенном периоде времени для обслуживания активных вызовов для последующего обновления данных заявки.
* В МИС МО назначается врач для обслуживания активных вызовов. МИС МО передает данные о назначенном враче для последующего обновления данных заявки.
* В МИС МО фиксируется результат обслуживания активных вызовов. МИС МО передает данные результата для последующего обновления данных заявки.
* В МИС МО фиксируется отказ в обслуживании. МИС МО передает данные об отказе для последующего обновления данных заявки.
* МИС СМП со своей стороны запрашивает список заявок на обслуживание активных вызовов для просмотра результатов обслуживаний.

Схема процесса обмена данными представлена ниже:

Рисунок 1 Схема процесса обмена заявками на обслуживание активного вызова



Поддержка процесса обмена информацией по заявкам на обслуживание активных вызовов между медицинскими системами МО является основным назначением сервиса Активы. В это входит:

* Получение и хранение, обновление заявок на обслуживание активных вызовов.
* Получение и хранение данных о пациентах, которым потребовался активный вызов.
* Получение и хранение данных о медперсонале, задействованном в процессе обслуживания вызовов.
* Передача по запросу заявок на обслуживание активных вызовов и передача их по запросу.
* Передача по запросу результатов обслуживания активных вызовов.

Для заявок на обслуживание активных вызовов в роли направляющей МО выступает СМП, в роли целевой – МО амбулаторного профиля.

Реализованное решение определено, исходя из требований бизнес-процесса обслуживания активного вызова и на основе документа статистического учета «Учетная форма N 110/у «Карта вызова скорой медицинской помощи»».

## Описание взаимодействия с сервисом

Сервис Активы предназначен для получения, ведения, хранения, поиска и выдачи сведений заявок на обслуживание активных вызовов МО в рамках региона.

Обмен данными между МИС СМП, МИС МО и сервиса Активы осуществляется в результате выполнения шагов следующего базового сценария:

1. Создание заявки на обслуживание активного вызова. Заявка из МИС СМП передается в сервис Активы.
2. Редактирование заявки. Передача изменений заявки из МИС СМП в сервис Активы.
3. Запрос списка заявок на обслуживание активных вызовов для действий. МИС МО периодически запрашивает наличие заявок у сервиса Активы для последующих действий.
4. Получение контекста по заявке. МИС МО запрашивает объект контекста заявки по ее идентификатору.
5. Отказ в обслуживании. Передача информации из МИС МО в сервис Активы о невозможности обслуживания активного вызова.
6. Назначение времени обслуживания активного вызова в МО. Информация о предварительно зарезервированном периоде времени для обслуживания активного вызова передается из МИС МО в сервис Активы.
7. Назначение врача для обслуживания активного вызова в МО. Информация о назначенном враче для обслуживания активного вызова передается из МИС МО в сервис Активы.
8. Передача результата обслуживания. Результат обслуживания активного вызова передается из МИС МО в сервис Активы
9. Запрос списка заявок на обслуживание активных вызов для просмотра результата. МИС СМП запрашивает список заявок у сервиса Активы для просмотра результатов обслуживания.

Описание сценариев взаимодействия с сервисом Активы на рисунке **Рисунок 2 Описание сценариев взаимодействия пользователя с сервисом Активы**.

Рисунок 2 Описание сценариев взаимодействия пользователя с сервисом Активы



# Описание протокола взаимодействия

## Общая информация о сервисе

В качестве протокола взаимодействия используется RESTful API. Данные необходимо передавать в формате JSON, должен присутствовать http заголовок content-type: application/json.

Информационный обмен может осуществляется в соответствии со стандартом FHIR® (Fast Healthcare Interoperability Resources), разработанным организацией HL7. Требуемая версия FHIR R4, 4.0.0. Подробное описание стандарта — <http://hl7.org/fhir/>. Использование REST- протокола в FHIR® – см. <http://fhir-ru.github.io/http.html>.

## Использование справочников

Справочники, используемые в сервисе, опубликованы в «Сервисе Терминологии». Описание сервиса Терминологии и правила взаимодействия с ним приведены по ссылке: http://api.netrika.ru/docs.php?article=Terminology.

Для каждого использованного в обмене справочника в схеме данных указан OID (объектный идентификатор).

Параметры схемы, указывающие на значения справочников, представляются в следующей виде:

{

"type": "string",

"title": "<Текстовое значение параметра>",

"description": "<Инструкции по заполнению с указанием источника - oid справочника>"

}

Роли пользователей передаются согласно справочнику НСИ: 1.2.643.5.1.13.2.1.1.734

Медицинские организации передаются согласно справочнику НСИ: 1.2.643.2.69.1.1.1.64

Полный перечень используемых сервисом справочников перечислен в описании параметров объектов.

## 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.

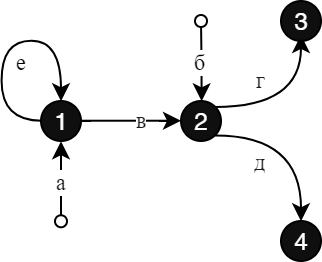
Данные, используемые при операциях с заявками - json-схема, описывающая набор и структуру данных необходимых для передачи со стороны пользователя при совершении операции с заявкой. Можно создать в рамках предметной области столько, сколько требуется.

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.

Рисунок 3 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).

Для сервиса Активы определен следующий набор состояний заявки:

* Черновик
* Направлено в МО
* Отказано в обслуживании
* Отменено
* Назначено время посещения
* Передано врачу
* Посещение не удалось
* Посещение прошло успешно

Каждое из состоний заявки имеет своой уникальный идентификатор в системе.

Для сервиса Активы определен следующий набор операций для заявки:

Таблица 2 Операции с заявкой в сервисе Активы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название** | **Начальное состояние** | **Итоговое состояние** | **Доступность операции для действия** | **Доступность для просмотра (по конечному состоянию)** |
| Создание заявки |  | Черновик | Все роли | Все роли |
| Редактирование черновика | Черновик | Черновик | Доступно специалистам СМП | [Доступно специалистам СМП](http://r75-rc.zdrav.netrika.ru/tm.admin/validators/6790c352-5e6d-4d88-b630-73c5cc92e4a3) |
| Направить в МО | Черновик | Направлено в МО | Доступно специалистам СМП | Доступно ответственным специалистам СМП и диспетчеру целевой МО |
| Отклонить по формальному признаку | Направлено в МО | Отменено | Доступно всем специалистам целевой МО | Доступно ответственным специалистам СМП и диспетчеру целевой МО |
| Назначить время посещения врача | Направлено в МО | Назначено время посещения | Доступно всем специалистам целевой МО | Доступно ответственным специалистам СМП и целевой МО |
| Передать врачу | Назначено время посещения | Передано врачу | Доступно всем специалистам целевой МО | Доступно ответственным специалистам СМП и целевой МО |
| Отменить | Назначено время посещения | Отменено | Доступно всем специалистам целевой МО | Доступно ответственным специалистам СМП и диспетчеру целевой МО |
| Посещение прошло успешно | Передано врачу | Посещение прошло успешно | Доступно всем специалистам целевой МО | Доступно ответственным специалистам СМП и целевой МО |
| Посещение не удалось | Передано врачу | Посещение не удалось | Доступно всем специалистам целевой МО | Доступно ответственным специалистам СМП и целевой МО |

### 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». Например, для предметной области «Тестовая область» 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 в последствии можно запросить методом «getSchema».

# Описание методов

## Состав методов сервиса

Функциональность сервиса Активы обеспечивается следующими методами:

* Создание заявки (POST //StartNewProcess), для стандарта FHIR (POST // api/Fhir/StartNewProcess).
* Редактирование заявки (POST //MoveToStage), для стандарта FHIR (POST // api/Fhir/MoveToStage).
* Получение списка доступных для действия заявок (POST // GetTransitionAvailableProcesses).
* Отказать в обслуживании (POST //MoveToStage), для стандарта FHIR (POST // api/Fhir/MoveToStage).
* Получение объекта контекста заявки (POST //GetProcessContext), для стандарта FHIR (POST //api/Fhir/ProcessContext).
* Получение маршрута обработки заявки по идентификатору (GetWorkflow).
* Получение схемы данных передаваемых при осуществлении перехода (GetSchema).
* Назначение времени посещения (POST //MoveToStages), для стандарта FHIR (POST //api/Fhir/MoveToStage).
* Назначение врача (POST //MoveToStage), для стандарта FHIR (POST //api/Fhir/MoveToStage).
* Передача результата обслуживания (POST //MoveToStage), для стандарта FHIR (POST //api/Fhir/MoveToStage).
* Получение списка доступных для просмотра заявок (POST //GetReadAvailableProcesses).
* Передача файла вложения заявки (POST //api/Commands/xds).
* Получение файла вложения заявки (POST //xds).

## Структура данных заявки

В методах сервиса для передачи набора данных заявки на обслуживание активного вызова используется параметр типа «Object» - processContext.

В обмене для структуры данных о заявке на обслуживание активного вызова по используется несколько объектов. Перечень объектов и описание их назначения представлено в таблице ниже.

Данные в параметре типа «Object» - processContext могут быть описаны в формате стандарту FHIR.

Таблица 3 Описание объектов, входящих в состав заявки

| **№ п/п** | **Объект** | **Нназначение объекта по бизнес-логике** | **Описание** |
| --- | --- | --- | --- |
| 1 | patient | Пациент | В объекте указывается информация о пациенте. |
| 2 | \_Role | dispatcherRole - роль диспетчера | В объекте указывается информация о ролевой принадлежности специалиста обслуживающего вызов |
|  |  | paramedicRole - роль фельдшера скорой помощи |  |
|  |  | doctorRole - роль врача, назначенного для обслуживания активного вызова в МО |  |
|  |  | seniorParamedicRole - роль старшего смены станции СМП |  |
|  |  | depChiefRole - роль руководителя СМП |  |
| 3 | condition | Диагноз | В объекте указывается информация об анамнезе и диагнозах установленных при осмотре пациента на вызове. |
| 4 | serviceRequest | Описание вызова | В объекте указывается информация об условиях и параметрах вызова |
| 5 | observation | Объективные данные осмотра | В объекте указывается информация о состоянии пациента, объективные данные о состоянии пациента при осмотре |
| 6 | attachedfiles | Вложения | В объекте указывается информация о вложениях и подписях мед.сотрудников, участвующих в процессе обработки вызова |
| 7 | appointment | Бронирование времени обслуживания | В объекте указывается информация о зарезервированном времени для обслуживания активного вызова в МО |
| 8 | appointmentResponse | Результат обслуживания | В объекте указывается информация о результате обслуживания активного вызова врачом МО |
| 9 | communication | Отказ в обслуживании | В объекте указывается информация об отказ в обслуживании |

## Структура метаданных заявки

Использование метаданных заявок применяется для вывода контекстной информацию о заявках в списке, для построения форм фильтрации по заявкам. Это общая логика для всех заявок вне зависимости от методов работы с заявками.

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).

"schema":

{

   "type": "object",

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

   "properties": {

     "patient": {

        "selector": "$.patient.idMPI"

                },

     "performer": {

                    "selector": "$.serviceRequest.performerOrganization"

                },

      "requester": {

         "selector": "$.serviceRequest.requesterOrganization"

                },

      "resultMedicalCare": {

         "selector": "$.serviceRequest.resultMedicalCare"

                },

      "resultAmbulanceDepartureType": {

         "selector": "$.serviceRequest.resultAmbulanceDepartureType"

                }

            }

}

Пример структуры метаданных конкретной заявки по схеме, приведенной выше:

"metadata":

{

"patient": "8ff30a0b-85c3-462c-aae1-3ec719b3c1a3",

"performer": "fc2c38ce-6599-4ff3-ae82-915b91a07db9",

"resultMedicalCare": "5",

"resultAmbulanceDepartureType": "3"

},

* 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).

**Описание принципа сопоставления Json-объектов при использовании 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 в формате обмена:

{

  "patientName": "Alexander",

  "diagnosis": {

    "name": "some name",

    "code": 315

  },

  "participant": {

    "name": "Ivan",

    "cpecialityCodes": [

      103,

      203,

      304

    ]

  }

}

В таком случае, 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). То 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 структура, представляющая собой массив дат, сериализованный в JSON:

"created": [

    "2020-12-17T11:58:30+03:00",

    "2020-12-18T13:58:30+03:00"

        ]

или

"updated": [

    "2020-12-17T11:58:30+03:00",

    "2020-12-18T13:58:30+03:00"

        ]

В случае если в массиве одна дата, будут выбраны все данные, в которых искомая дата будет больше чем, переданная в дата массиве. 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.

## Описание параметров объектов, входящих в структуру заявки

**Описание параметров Patient**

Перечень параметров объекта Patient и их описание представлены в таблице ниже.

Таблица 4 Описание параметров объекта Patient

| **Параметры** | **Тип** | **Обязательность** | **Описание** |
| --- | --- | --- | --- |
| patient: |  |  |  |
| idMPI | guid | 1..1 | Идентификатор пациента из подсистемы MPI |
| identityDocument/ | array |  | Массив данных документов, удостоверяющих личность |
| /id | string | 0..1 | Идентификатор документа |
| /system | string | 0..1 | Тип документа, удостоверяющего пациента, по справочнику НСИ 1.2.643.2.69.1.1.1.6 |
| /code | string | 0..1 | Номер документа, удостоверяющего личность пациента |
| /id | string | 0..1 | Идентификатор документа |
| /system | string | 0..1 | Тип документа пациента, по справочнику НСИ 1.2.643.2.69.1.1.1.6.228 – Полис ОМС |
| /code | int | 0..1 | Номер полиса |
| birthDate | date | 0..1 | Дата рождения |
| gender | string | 0..1 | Пол |
| social/ |  |  | Данные социального статуса |
| /registration | int | 0..1 | Тип регистрации, по справочнику НСИ 1.2.643.2.69.1.1.1.182.1 |
| /socialStatus | int | 0..1 | Социальный статус, по справочнику НСИ 1.2.643.2.69.1.1.1.182.2 |
| placeOfWork | string | 0..1 | Место работы |
| address/ |  | 0..1 | Данные адреса пациента |
| /district | string | 0..1 | Район |
| /city | string | 0..1 | Город |
| /line | string | 0..1 | Улица , номер дома |
| fullName | string | 0..1 | Фамилия Имя Отчество пациента |
| contact/ | string | 0..1 | Данные контактного лица, осуществившего вызов |
| /name | string | 0..1 | Фамилия Имя Отчество звонившего |
| /telecom | string | 0..1 | Телефон звонившего |

**Описание параметров объекта \_Role**

Перечень параметров объектов \_Role и их описание представлены в таблице ниже. Объекты с данными с о медицинских сотрудниках, участвующих в процессе обслуживания вызова: dispatcherRole, paramedicRole, seniorParamedicRole, depChiefRole, doctorRole.

Таблица 5 Описание параметров объектов \_Role

|  |  |  |  |
| --- | --- | --- | --- |
| **Параметры** | **Тип** | **Обязательность** | **Описание** |
| \_Role: |  |  |  |
| identityDocument/ | string | 1..1 | Данные документов мед.сотрудника |
| /system | string | 1..1 | Тип документа мед.сотрудника, по справочнику НСИ 1.2.643.2.69.1.1.1.6.228 – СНИЛС |
| /code | string | 1..1 | Номер документа |
| dispatcherNumber | string | 0..1 | Табельный номер, передается для роли диспетчера, в других случаях не передается |
| organization | guid | 0..1 | Идентификатор МО, по справочнику |

**Описание параметров объекта ServiceRequest**

Перечень параметров объекта ServiceRequest и их описание представлены в таблице ниже.

Таблица 6 Описание параметров объекта ServiceRequest

| **Параметры** | **Тип** | **Обязательность** | **Описание** |
| --- | --- | --- | --- |
| serviceRequest: |  |  |  |
| reasonCallMedicalCare | int | 0..1 | Повод к вызову, по справочнику НСИ 1.2.643.2.69.1.1.1.182.3 |
| territoryCode | int | 0..1 | Код территории |
| requesterOrganization | string | 0..1 | Идентификатор направляющей МО по справочнику НСИ 1.2.643.2.69.1.1.1.64, Номер станции скорой помощи, |
| shiftNumber | string | 0..1 | Номер смены |
| brigadeNumber | string | 0..1 | Номер бригады |
| ambulanceTeam | int | 0..1 | Состав бригады скорой медицинской помощи, по справочнику НСИ 1.2.643.2.69.1.1.1.182.7 |
| ambulanceCallType | int | 0..1 | Тип вызова: перв/повт, по справочнику НСИ 1.2.643.2.69.1.1.1.182.4 |
| placeCallReceived | int | 0..1 | Место получения вызова бригадой скорой медицинской, по справочнику НСИ 1.2.643.2.69.1.1.1.182.5 |
| reasonsBeingLate | int | 0..1 | Причины выезда с опозданием, по справочнику НСИ 1.2.643.2.69.1.1.1.182.6 |
| accidentCause | int | 0..1 | Причина несчастного случая, по справочнику НСИ 1.2.643.2.69.1.1.1.182.9 |
| intoxication | boolean | 0..1 | Наличие опьянения (да/нет) |
| complaint | string | 0..1 | Описание жалоб пациента |
| complicationsExamination/ |  | 0..1 | Данные об осложнениях при осмотре бригады скорой помощи |
| /complications | int | 0..1 | Осложнения при осмотре бригады скорой помощи, по справочнику НСИ 1.2.643.2.69.1.1.1.182.23 |
| /complicationsCodMKB | string | 0..1 | Код диагноза для осложнения, по справочнику НСИ 1.2.643.2.69.1.1.1.2 (МКБ-10) |
| /effectivenessMeasures | int | 0..1 | Эффективность мероприятий бригады скорой помощи, по справочнику НСИ 1.2.643.2.69.1.1.1.182.24 |
| /assistanceProvidedLocationCallАmbulance | string | 0..1 | Описание оказанной помощи на месте вызова. |
| /assistanceProvidedАmbulanceCar | string | 0..1 | Описание оказанной помощи в автомобиле скорой. |
| performanceIndicatorsEvents/ |  | 0..1 | Показатели эффективности мероприятий бригады скорой. Заполняется по справочнику НСИ 1.2.643.2.69.1.1.1.182.28 |
| /bloodPressure | string | 0..1 | Данные измерений АД |
| /pulse | string | 0..1 | Данные измерений пульса |
| /heartRate | string | 0..1 | Данные измерений ЧСС |
| /respirationRate | string | 0..1 | Данные измерений ЧД |
| /bodyHeat | string | 0..1 | Данные измерений температуры тела |
| /pulseOximetry | string | 0..1 | Данные измерений пульсометрии |
| /glucose monitoring/ | string | 0..1 | Данные измерений глюкометрии |
| /bloodGlucoseLowerBound | string | 0..1 | Нижняя граница уровня глюкозы в крови |
| /bloodGlucoseUpperBound | string | 0..1 | Верхняя граница уровня глюкозы в крови |
| consent | boolean | 0..1 | Указатель согласия пациента |
| time/ |  | 0..1 | Данные о времени обработки вызова |
| /receiveСallDateTime | dateTime | 0..1 | Дата время приема вызова диспетчером СМП, в формате 2010-10-07T09:41:23.507+00:00 |
| /callTransferDateTime | dateTime | 0..1 | Время передачи бригаде, в формате 2010-10-08T09:43:23.507+00:00 |
| /checkoutTime | dateTime | 0..1 | Время выезда на вызов, в формате hh:mm:ss |
| /arrivalTime | dateTime | 0..1 | Время прибытия на место вызова, в формате hh:mm:ss |
| /startTimeTransportation | dateTime | 0..1 | Время начала транспортировки, в формате hh:mm:ss |
| /arrivalTimeСlinic | dateTime | 0..1 | Время прибытия в МО, в формате hh:mm:ss |
| /endTimeCall | dateTime | 0..1 | Время окончания вызова, в формате hh:mm:ss |
| /timeReturnToStation | dateTime | 0..1 | Время возвращения бригады СМП на станцию, в формате hh:mm:ss |
| /timeTakenComplete | string | 0..1 | Общее затраченное на вызов время |
| kilometers | string | 0..1 | Километраж (км) |
| resultMedicalCare | int | 0..1 | Результат оказания скорой медицинской помощи. Значение из справочника НСИ 1.2.643.2.69.1.1.1.182.25 |
| performerOrganization | guid | 0..1 | Идентификатор МО, для которой передается уведомление о вызове (МО по прикреплению пациента), по справочнику НСИ 1.2.643.2.69.1.1.1.64 |
| deliveryMethodToCar | int | 0..1 | Способ доставки больного в автомобиль скорой , по справочнику НСИ 1.2.643.2.69.1.1.1.182.26 |
| resultAmbulanceDepartureType | int | 0..1 | Тип результата выезда бригады скорой помощи, по справочнику НСИ 1.2.643.2.69.1.1.1.182.27 |
| hospitalOrganization | guid | 0..1 | Идентификатор МО по справочнику НСИ 1.2.643.2.69.1.1.1.64, передается в случае госпитализации пациента |
| idIEMK | string | 0..1 | Идентификатор случая в подсистеме ИЕМК |
| comments | text | 0..1 | Примечания по вызову |

**Описание параметров объекта Observation**

Перечень параметров объекта Observation и их описание представлены в таблице ниже.

Таблица 7 Описание параметров объекта Observation

| **Параметры** | **Тип** | **Обязательность** | **Описание** |
| --- | --- | --- | --- |
| observation: |  |  |  |
| generalCondition | int | 0..1 | Общее состояние больного. Значение из справочника НСИ 1.2.643.2.69.1.1.1.182.10 |
| behavior | int | 0..1 | Поведение больного. Значение по справочнику НСИ1.2.643.2.69.1.1.1.182.11 |
| consciousness | int | 0..1 | Сознание больного. Значение по справочнику НСИ 1.2.643.2.69.1.1.1.182.12 |
| meningealSigns | int | 0..1 | Менингеальные знаки. Значение по справочнику НСИ 1.2.643.2.69.1.1.1.182.12 |
| eyes/ |  | 0..1 | Данные осмотра состояния зрачков. Значение по справочнику НСИ 1.2.643.2.69.1.1.1.182.13 |
| /pupils | int | 0..1 | Зрачки |
| /anisocoria | int | 0..1 | Анизокория |
| /nystagmus | int | 0..1 | Нистагматизм |
| /lightResponse | int | 0..1 | Реакция на свет |
| skin/ |  | 0..1 | Данные осмотра состояния кожных покровов. Значение по справочнику НСИ 1.2.643.2.69.1.1.1.182.14 |
| /integument | int | 0..1 | Кожные покровы |
| /acrocyanosis | int | 0..1 | Акроцианоз |
| /marbling | int | 0..1 | Мраморность |
| /edemas | int | 0..1 | Отеки |
| /localizationOfEdema: | text | 0..1 | Описание локализации отеков |
| /rash | int | 0..1 | Сыпь |
| /rashLocalization | text | 0..1 | Описание локализации сыпи |
| breathing | int | 0..1 | Дыхание. Значение по справочнику НСИ 1.2.643.2.69.1.1.1.182.15 |
| wheezing | int | 0..1 | Хрипы. Значение по справочнику НСИ 1.2.643.2.69.1.1.1.182.16 |
| shortnessOfBreath | int | 0..1 | Одышка. Значение по справочнику НСИ 1.2.643.2.69.1.1.1.182.17 |
| organsCirculatorySystem/ |  | 0..1 | Органы системы кровообращения |
| /heartSounds | int | 0..1 | Тоны сердца. Значение по справочнику НСИ 1.2.643.2.69.1.1.1.182.18 |
| /noise | int | 0..1 | Шум сердца. Значение по справочнику НСИ 1.2.643.2.69.1.1.1.182.19 |
| /pulseLike | int | 0..1 | Описание пульса. Значение по справочнику НСИ 1.2.643.2.69.1.1.1.182.20 |
| digestiveOrgans/ |  | 0..1 | Органы пищеварения |
| /tongue | int | 0..1 | Язык. Значение по справочнику НСИ 1.2.643.2.69.1.1.1.182.21 |
| /belly | int | 0..1 | Живот. Значение по справочнику НСИ 1.2.643.2.69.1.1.1.182.22 |
| /bellyInActOfBreathing | int | 0..1 | Живот: участвует в акте дыхания. Значение по справочнику НСИ 1.2.643.2.69.1.1.1.182.22 |
| /peritonealIrritationSymptoms | int | 0..1 | Симптомы раздражения брюшины. Значение по справочнику НСИ 1.2.643.2.69.1.1.1.182.22 |
| /enlargedLiver | int | 0..1 | Печень. Значение по справочнику НСИ 1.2.643.2.69.1.1.1.182.22 |
| urination | text | 0..1 | Мочеиспускание |
| feces | text | 0..1 | Стул |
| otherSymptoms | text | 0..1 | Описание других симптомов |
| pressure/ |  | 0..1 | Показатели АД |
| /workingBloodPressure | string | 0..1 | Рабочее АД |
| /bloodPressure | string | 0..1 | Данные измерений АД |
| pulse | string | 0..1 | Данные измерений пульса |
| heartRate | string | 0..1 | Данные измерений ЧСС |
| respirationRate | string | 0..1 | Данные измерений ЧД |
| bodyHeat | string | 0..1 | Данные измерений температуры тела |
| pulseOximetry | string | 0..1 | Данные измерений пульсометрии |
| glucose monitoring/ |  | 0..1 | Данные измерений глюкометрии |
| /bloodGlucoseLowerBound | string | 0..1 | Нижняя граница уровня глюкозы в крови |
| /bloodGlucoseUpperBound | string | 0..1 | Верхняя граница уровня глюкозы в крови |
| additionalObjectiveData | text | 0..1 | Дополнительные объективные данные. Локальный статус. |
| ECG/ |  | 0..1 | Электрокардиограмма (ЭКГ) |
| /preMedicalCareECG | text | 0..1 | Описание измерений ЭКГ до оказания медицинской помощи |
| /preMedicalCareECGTime | time | 0..1 | Время проведения |
| /postMedicalCareECG: | text | 0..1 | Описание измерений ЭКГ после оказания медицинской помощи |
| /postMedicalCareECGTime | time | 0..1 | Время проведения |

**Описание параметров объекта Condition**

Перечень параметров объекта Condition и их описание представлены в таблице ниже.

Таблица 8 Описание параметров объекта Condition

| **Параметры** | **Тип** | **Обязательность** | **Описание** |
| --- | --- | --- | --- |
| condition: |  |  |  |
| codeMKB | string | 1..1 | Код диагноза по справочнику НСИ 1.2.643.2.69.1.1.1.2 (МКБ-10) |
| complaints | text | 0..1 | Жалобы |
| anamnesis | text | 0..1 | Дополнительные сведения по анамнезу заболевания |

**Описание параметров объекта attachedfiles**

Перечень параметров объекта attachedfiles и их описание представлены в таблице ниже. Объект предназначен для обмена данными вложений: файлы документов, измерений (ЭКГ), ЭЦП.

Таблица 9 Описание параметров объекта attachedfiles

| **Параметры** | **Тип** | **Обязательность** | **Описание** |
| --- | --- | --- | --- |
| attachedfiles | array | 0..1 |  |
| id | int | 1..1 | Идентификатор вложения |
| fileURL | string | 0..1 | Ссылка на файл |
| signatureURL | string | 0..1 | Ссылка на подпись |
| isBlocked | boolean | 0..1 | Логическое удаление |
| 110u/ |  |  |  |
| /fileURL | string | 0..1 | Ссылка на файл формы 110у |
| /signatureURL | string | 0..1 | Ссылка на подпись ЭЦП |

**Описание параметров объекта Appointment**

Перечень параметров объекта Appointment и их описание представлены в таблице ниже.

Таблица 10 Описание параметров объекта Appointment

| **Параметры** | **Тип** | **Обязательность** | **Описание** |
| --- | --- | --- | --- |
| appointment: |  |  |  |
| start | dateTime | 0..1 | Желаемые дата и время обслуживания, начало периода, в формате 2020-10-26T10:00:00Z |
| end | dateTime | 0..1 | Желаемые дата и время обслуживания, окончание периода, в формате 2020-10-26T10:00:00Z |
| requestedPeriod/ |  | 0..1 | Забронированный период времени для обслуживания активного вызова |
| /start | dateTime | 0..1 | Забронированное дата и время обслуживания, начало периода, в формате 2020-10-26T10:00:00Z |
| /end | dateTime | 0..1 | Забронированное дата и время обслуживания, окончание периода, в формате 2020-10-26T10:00:00Z |

**Описание параметров объекта AppointmentResponse**

Перечень параметров объекта AppointmentResponse и их описание представлены в таблице ниже.

Таблица 11 Описание параметров объекта AppointmentResponse

| **Параметры** | **Тип** | **Обязательность** | **Описание** |
| --- | --- | --- | --- |
| appointmentResponse/ |  |  |  |
| /start | dateTime | 0..1 | Дата и время начала фактического обслуживания активного вызова, в формате 2020-10-26T10:00:00Z |
| /end | dateTime | 0..1 | Дата и время окончания фактического обслуживания активного вызова, в формате 2020-10-27T10:00:00Z |
| comment | text | 0..1 | Описание результата. Текстовый комментарий по результату обслуживания |

## Создание заявки (POST StartNewProcess)

Для создания заявки об активном вызове в сервисе используется метод POST StartNewProcess. Метод создаёт заявку по маршруту согласно переданному переходу.

### **Описание параметров запроса**

В таблице ниже представлено описание параметров запроса метода.

Таблица 12 Входные параметры для метода POST StartNewProcess

| **№ 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 (структура, смотреть в контрольных примерах) |
| 5 | roleContext | Object | 1..1 | Значение ролевого контекста пользователя |

### Пример запроса

POST {{url}}/api/Commands/StartNewProcess

{

  "workflowId": "b3393395-06f7-4261-b403-5dc0dab1296b",

  "name": "Моя заявка",

  "initialTransitionId": "e9b2cce6-93ec-4118-8728-1580af7b9e82",

  "processContext": { <структура, смотреть в контрольных примерах> },

  "roleContext": [ .. ]

}

### **Описание параметров ответа**

В таблице ниже представлено описание параметров ответа метода.

Таблица 13 Выходные параметры для метода POST StartNewProcess

| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| --- | --- | --- | --- | --- |
| 1 | workflowId | String | 0..1 | Идентификатор маршрута |
| 2 | processId | String | 0..1 | Идентификатор созданного процесса по заявке |
| 3 | stageId | String | 0..1 | Идентификатор статуса заявки |
| 4 | currentTransition | String | 0..1 | Идентификатор текущего перехода (для создания заявки значение null) |
| 5 | humanFriendlyId | String | 0..1 | Адаптированный идентификатор заявки для использования в обмене внешних систем |
| 6 | validationResults | String | 0..1 | Результат валидации |
| 7 | success | String | 0..1 | Результат выполнения |
| 8 | errorCode | String | 0..1 | Код ошибки |
| 9 | message | String | 0..1 | Описание ошибки |
| 10 | stackTrace | String | 0..1 | Служебный параметр |

### Пример ответа

{

"workflowId": "5fb7cefc-b7e0-467c-b79b-43f2859c95dc",

"processId": "11115d34-38b5-48be-b419-0bf2438c0ae2",

"stageId": "617690fd-de03-41d6-b2df-793f765ef537",

"currentTransition": null,

"humanFriendlyId": "TMC0621Y98CD7",

"validationResults": null,

"success": true,

"errorCode": 0,

"message": null,

"stackTrace": null

}

## Редактирование/обновление заявки (POST MoveToStage)

Для редактирования или обновления данных заявки об активном вызове в сервисе используется метод POST MoveToStage. В базовом назначении метод позволяет осуществить запрос перехода заявки в новое состояние согласно указанному правилу перехода. Далее представлены варианты использования запроса метода при которых происходит переход заявки в новое состояние.

Помимо редактирования заявки, запрос метода используется для обновления состояния заявки в следующих случаях:

* Направление в МО для назначения времени посещения.
* Назначение врача.
* Передача результата обслуживания.
* Отказ в обслуживании на разных этапах обработки заявки.

### **Описание параметров запроса**

В таблице ниже представлено описание параметров запроса метода.

Таблица 14 Входные параметры для метода POST MoveToStage

| **№ 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 (структура, смотреть в контрольных примерах) |
| 4 | roleContext | Object | 1..1 | Значение ролевого контекста пользователя |

### Примеры запроса

POST {{url}}/api/Commands/MoveToStage

{

"processId": "b26c9527-c67b-4702-a525-df20a0645d23",

"transitionId": "6545013e-c9bc-4e35-b466-802769a394bc",

"processContext":{<структура, смотреть в контрольных примерах>},

"roleContext": [ ..]

}

### **Описание параметров ответа**

В таблице ниже представлено описание параметров ответа метода.

Таблица 15 Выходные параметры для метода POST MoveToStage

| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| --- | --- | --- | --- | --- |
| 1 | workflowId | String | 0..1 | Идентификатор маршрута |
| 2 | processId | String | 0..1 | Идентификатор созданного процесса по заявке |
| 3 | stageId | String | 0..1 | Идентификатор статуса заявки |
| 4 | currentTransition | String | 0..1 | Идентификатор текущего перехода (для создания заявки значение null) |
| 5 | humanFriendlyId | String | 0..1 | Адаптированный идентификатор заявки для использования в обмене внешних систем |
| 6 | validationResults | String | 0..1 | Результат валидации |
| 7 | success | String | 0..1 | Результат выполнения |
| 8 | errorCode | String | 0..1 | Код ошибки |
| 9 | message | String | 0..1 | Описание ошибки |
| 10 | stackTrace | String | 0..1 | Служебный параметр |

### Примеры ответа

{

"workflowId": "5fb7cefc-b7e0-467c-b79b-43f2859c95dc",

"processId": "d456aa53-4ae8-4599-b0a2-655043d1ec1c",

"stageId": "54a9b8d5-24b9-454c-b197-635aeb963311",

"currentTransition": null,

"humanFriendlyId": "TMC06218Q9IA3",

"validationResults": null,

"success": true,

"errorCode": 0,

"message": null,

"stackTrace": null

}

## Получение списка доступных для действия заявок (POST //GetTransitionAvailableProcesses)

Для получения списка доступных для действия заявок в сервисе используется метод POST GetTransitionAvailableProcesses. Метод предназначен для получения списка заявок в статусах, которые имеют операции доступные для использования данному пользователю.

### **Описание параметров запроса**

В таблице ниже представлено описание параметров запроса метода.

Таблица 17 Входные параметры для метода фильтрации

| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| --- | --- | --- | --- | --- |
| 1 | roleContext | String | 1..1 | Значение ролевого контекста пользователя.  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). |
| 2 | workflowFilter | Object | 0...1 | Параметры фильтра по схеме описания метаданных маршрута.  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 под таблицей) |
| 3 | processFilter | Object | 0...1 | Параметры фильтра по схеме описания метаданных заявки.  ProcessFilter – фильтр заявок непосредственно по атрибутам и матаданным 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). |
| 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 |
| 5 | Take | Number | 0..1 | Количество элементов в списке |

Подробное описание и примеры структуры схем метаданных для последующего использования в методах фильтрации заявок представлено в пункте 5.3.

Таблица 18 Выходные параметры для метода фильтрации

| **№ 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 запроса

POST {{url}}/api/Queries/GetTransitionAvailableProcesses

{

    "RoleContext": [

        {

            "Role": "DEPCHIEF",

            "Organization": "931a9317-586c-4dd5-bc32-cd8d3af78903"

        }

    ],

    "WorkflowFilter": {},

    "ProcessFilter": {

        "workflow": "b3393395-06f7-4261-b403-5dc0dab1296b",

        "metadata": {

            "resultMedicalCare": [

                "3",

                "5"

            ],

            "resultAmbulanceDepartureType": [

                "3",

                "1"

            ]

        },

        "updated": [],

        "created": [

            "2020-12-17T11:58:30+03:00",

            "2020-12-18T13:58:30+03:00"

        ]

    },

    "orderingField": "updated",

    "descendingOrder": false,

    "BusinessStatusCodes": [

        "2"

    ],

    "StageFilter": [],

    "Skip": 0,

    "Take": 10

}

### Пример ответа

{

    "result": {

        "result": [

            {

                "transitionIds": [

                    "6545013e-c9bc-4e35-b466-802769a394bc",

                    "959450aa-a67d-45e7-b2ba-2f682959c12d"

                ],

                "processId": "3c9b3659-b2b3-424a-ab87-cfde37f1196e",

                "processHumanFriendlyId": "TMC122043OOX5",

                "metadata": {

                    "patient": "8ff30a0b-85c3-462c-aae1-3ec719b3c1a3",

                    "performer": "fc2c38ce-6599-4ff3-ae82-915b91a07db9",

                    "resultMedicalCare": "5",

                    "resultAmbulanceDepartureType": "3"

                },

                "currentStageId": "d41a2f57-6c06-4f38-9982-ea1fc7730d01",

                "currentStage": "Черновик",

                "workflowId": "b3393395-06f7-4261-b403-5dc0dab1296b",

                "workflowName": "Активы\_бп",

                "processName": "Новая заявка актив",

                "created": "2020-12-17T13:33:18.966651+03:00",

                "updated": "2020-12-17T13:33:18.966651+03:00",

                "businessStatus": {

                    "system": "urn:oid:1.2.643.2.69.1.1.1.148.2",

                    "code": "2"

                }

            },

            {

                "transitionIds": [

                    "6545013e-c9bc-4e35-b466-802769a394bc",

                    "959450aa-a67d-45e7-b2ba-2f682959c12d"

                ],

                "processId": "2806edea-0c7f-4f4f-9c01-faa4761eb286",

                "processHumanFriendlyId": "TMC12200L75IN",

                "metadata": {

                    "patient": "8ff30a0b-85c3-462c-aae1-3ec719b3c1a3",

                    "performer": "fc2c38ce-6599-4ff3-ae82-915b91a07db9",

                    "resultMedicalCare": "5",

                    "resultAmbulanceDepartureType": "3"

                },

                "currentStageId": "d41a2f57-6c06-4f38-9982-ea1fc7730d01",

                "currentStage": "Черновик",

                "workflowId": "b3393395-06f7-4261-b403-5dc0dab1296b",

                "workflowName": "Активы\_бп",

                "processName": "Новая заявка актив",

                "created": "2020-12-17T15:33:08.284762+03:00",

                "updated": "2020-12-17T15:33:08.284762+03:00",

                "businessStatus": {

                    "system": "urn:oid:1.2.643.2.69.1.1.1.148.2",

                    "code": "2"

                }

            },

            {

                "transitionIds": [

                    "6545013e-c9bc-4e35-b466-802769a394bc",

                    "959450aa-a67d-45e7-b2ba-2f682959c12d"

                ],

                "processId": "5a6120a1-ceee-48c2-bd5c-a0c3cc83c980",

                "processHumanFriendlyId": "TMC122094XD5X",

                "metadata": {

                    "patient": "8ff30a0b-85c3-462c-aae1-3ec719b3c1a3",

                    "performer": "fc2c38ce-6599-4ff3-ae82-915b91a07db9",

                    "resultMedicalCare": "5",

                    "resultAmbulanceDepartureType": "3"

                },

                "currentStageId": "d41a2f57-6c06-4f38-9982-ea1fc7730d01",

                "currentStage": "Черновик",

                "workflowId": "b3393395-06f7-4261-b403-5dc0dab1296b",

                "workflowName": "Активы\_бп",

                "processName": "Новая заявка актив",

                "created": "2020-12-18T11:17:04.438225+03:00",

                "updated": "2020-12-18T11:17:04.438225+03:00",

                "businessStatus": {

                    "system": "urn:oid:1.2.643.2.69.1.1.1.148.2",

                    "code": "2"

                }

            },

            {

                "transitionIds": [

                    "6545013e-c9bc-4e35-b466-802769a394bc",

                    "959450aa-a67d-45e7-b2ba-2f682959c12d"

                ],

                "processId": "e90d54af-c1e8-4127-bb4f-f573d7d3353d",

                "processHumanFriendlyId": "TMC1220926SYY",

                "metadata": {

                    "patient": "8ff30a0b-85c3-462c-aae1-3ec719b3c1a3",

                    "performer": "fc2c38ce-6599-4ff3-ae82-915b91a07db9",

                    "resultMedicalCare": "5",

                    "resultAmbulanceDepartureType": "3"

                },

                "currentStageId": "d41a2f57-6c06-4f38-9982-ea1fc7730d01",

                "currentStage": "Черновик",

                "workflowId": "b3393395-06f7-4261-b403-5dc0dab1296b",

                "workflowName": "Активы\_бп",

                "processName": "Новая заявка актив",

                "created": "2020-12-18T11:17:12.202303+03:00",

                "updated": "2020-12-18T11:17:12.202303+03:00",

                "businessStatus": {

                    "system": "urn:oid:1.2.643.2.69.1.1.1.148.2",

                    "code": "2"

                }

            },

            {

                "transitionIds": [

                    "6545013e-c9bc-4e35-b466-802769a394bc",

                    "959450aa-a67d-45e7-b2ba-2f682959c12d"

                ],

                "processId": "4bf133a0-549f-4e28-b0e1-3a44c9b0bbde",

                "processHumanFriendlyId": "TMC122052YL1B",

                "metadata": {

                    "patient": "8ff30a0b-85c3-462c-aae1-3ec719b3c1a3",

                    "performer": "fc2c38ce-6599-4ff3-ae82-915b91a07db9",

                    "resultMedicalCare": "3",

                    "resultAmbulanceDepartureType": "3"

                },

                "currentStageId": "d41a2f57-6c06-4f38-9982-ea1fc7730d01",

                "currentStage": "Черновик",

                "workflowId": "b3393395-06f7-4261-b403-5dc0dab1296b",

                "workflowName": "Активы\_бп",

                "processName": "Новая заявка актив",

                "created": "2020-12-18T11:58:30.200089+03:00",

                "updated": "2020-12-18T11:58:30.200089+03:00",

                "businessStatus": {

                    "system": "urn:oid:1.2.643.2.69.1.1.1.148.2",

                    "code": "2"

                }

            },

            {

                "transitionIds": [

                    "6545013e-c9bc-4e35-b466-802769a394bc",

                    "959450aa-a67d-45e7-b2ba-2f682959c12d"

                ],

                "processId": "727523ee-059e-40bf-a396-abcff9160287",

                "processHumanFriendlyId": "TMC122055BU7D",

                "metadata": {

                    "patient": "8ff30a0b-85c3-462c-aae1-3ec719b3c1a3",

                    "performer": "fc2c38ce-6599-4ff3-ae82-915b91a07db9",

                    "requester": "931a9317-586c-4dd5-bc32-cd8d3af78903",

                    "resultMedicalCare": "3",

                    "resultAmbulanceDepartureType": "3"

                },

                "currentStageId": "d41a2f57-6c06-4f38-9982-ea1fc7730d01",

                "currentStage": "Черновик",

                "workflowId": "b3393395-06f7-4261-b403-5dc0dab1296b",

                "workflowName": "Активы\_бп",

                "processName": "Новая заявка актив",

                "created": "2020-12-18T12:23:30.272053+03:00",

                "updated": "2020-12-18T12:23:30.272053+03:00",

                "businessStatus": {

                    "system": "urn:oid:1.2.643.2.69.1.1.1.148.2",

                    "code": "2"

                }

            }

        ],

        "total": 6

    },

    "success": true,

    "errorCode": 0,

    "message": null,

    "stackTrace": null

}

## Получение списка доступных для просмотра заявок (POST //GetReadAvailableProcesses).

Для получения списка доступных для просмотра заявок в сервисе используется метод POST GetReadAvailableProcesses. Метод предназначен для получения списка заявок в статусах, которые доступны данному пользователю для просмотра и действия.

### **Описание параметров запроса**

В таблице ниже представлено описание параметров запроса метода.

Таблица 19 Входные параметры для метода фильтрации

| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| --- | --- | --- | --- | --- |
| 1 | roleContext | Object | 1..1 | Значение ролевого контекста пользователя.  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). |
| 2 | WorkflowFilter | Object | 0...1 | Параметры фильтра по схеме описания метаданных маршрута.  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) |
| 3 | ProcessFilter | Object | 0...1 | Параметры фильтра по схеме описания метаданных заявки.  ProcessFilter – фильтр заявок непосредственно по атрибутам и матаданным 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). |
| 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. Для сортировки по убыванию в поле запроса descendingOrder нужно передать true. |
| 7 | Skip | Number | 0..1 | Количество пропущенных элементов в выборке |
| 8 | Take | Number | 0..1 | Количество элементов в списке |

Подробное описание и примеры структуры схем метаданных для последующего использования в методах фильтрации заявок представлено в пункте 5.3.

### Пример запроса

POST {{url}}/api/Queries/GetReadAvailableProcesses

{

    "RoleContext": [

        {

            "Role": "DISPETCHER",

            "Organization": "931a9317-586c-4dd5-bc32-cd8d3af78903"

        }

    ],

    "WorkflowFilter": {},

    "ProcessFilter": {

        "workflow": "b3393395-06f7-4261-b403-5dc0dab1296b",

        "metadata": {

            "resultMedicalCare": [

                "3",

                "5"

            ],

            "resultAmbulanceDepartureType": [

                "3",

                "1"

            ]

        },

        "updated": [],

        "created": [

            "2020-12-17T11:58:30+03:00",

            "2020-12-18T13:58:30+03:00"

        ]

    },

    "orderingField": "updated",

    "descendingOrder": false,

    "BusinessStatusCodes": [

        "2"

    ],

    "StageFilter": [],

    "Skip": 0,

    "Take": 10

}

### Пример ответа

{

    "result": {

        "result": [

            {

                "processId": "3c9b3659-b2b3-424a-ab87-cfde37f1196e",

                "processHumanFriendlyId": "TMC122043OOX5",

                "metadata": {

                    "patient": "8ff30a0b-85c3-462c-aae1-3ec719b3c1a3",

                    "performer": "fc2c38ce-6599-4ff3-ae82-915b91a07db9",

                    "resultMedicalCare": "5",

                    "resultAmbulanceDepartureType": "3"

                },

                "currentStageId": "d41a2f57-6c06-4f38-9982-ea1fc7730d01",

                "currentStage": "Черновик",

                "workflowId": "b3393395-06f7-4261-b403-5dc0dab1296b",

                "workflowName": "Активы\_бп",

                "processName": "Новая заявка актив",

                "created": "2020-12-17T13:33:18.966651+03:00",

                "updated": "2020-12-17T13:33:18.966651+03:00",

                "businessStatus": {

                    "system": "urn:oid:1.2.643.2.69.1.1.1.148.2",

                    "code": "2"

                }

            },

            {

                "processId": "2806edea-0c7f-4f4f-9c01-faa4761eb286",

                "processHumanFriendlyId": "TMC12200L75IN",

                "metadata": {

                    "patient": "8ff30a0b-85c3-462c-aae1-3ec719b3c1a3",

                    "performer": "fc2c38ce-6599-4ff3-ae82-915b91a07db9",

                    "resultMedicalCare": "5",

                    "resultAmbulanceDepartureType": "3"

                },

                "currentStageId": "d41a2f57-6c06-4f38-9982-ea1fc7730d01",

                "currentStage": "Черновик",

                "workflowId": "b3393395-06f7-4261-b403-5dc0dab1296b",

                "workflowName": "Активы\_бп",

                "processName": "Новая заявка актив",

                "created": "2020-12-17T15:33:08.284762+03:00",

                "updated": "2020-12-17T15:33:08.284762+03:00",

                "businessStatus": {

                    "system": "urn:oid:1.2.643.2.69.1.1.1.148.2",

                    "code": "2"

                }

            },

            {

                "processId": "5a6120a1-ceee-48c2-bd5c-a0c3cc83c980",

                "processHumanFriendlyId": "TMC122094XD5X",

                "metadata": {

                    "patient": "8ff30a0b-85c3-462c-aae1-3ec719b3c1a3",

                    "performer": "fc2c38ce-6599-4ff3-ae82-915b91a07db9",

                    "resultMedicalCare": "5",

                    "resultAmbulanceDepartureType": "3"

                },

                "currentStageId": "d41a2f57-6c06-4f38-9982-ea1fc7730d01",

                "currentStage": "Черновик",

                "workflowId": "b3393395-06f7-4261-b403-5dc0dab1296b",

                "workflowName": "Активы\_бп",

                "processName": "Новая заявка актив",

                "created": "2020-12-18T11:17:04.438225+03:00",

                "updated": "2020-12-18T11:17:04.438225+03:00",

                "businessStatus": {

                    "system": "urn:oid:1.2.643.2.69.1.1.1.148.2",

                    "code": "2"

                }

            },

            {

                "processId": "e90d54af-c1e8-4127-bb4f-f573d7d3353d",

                "processHumanFriendlyId": "TMC1220926SYY",

                "metadata": {

                    "patient": "8ff30a0b-85c3-462c-aae1-3ec719b3c1a3",

                    "performer": "fc2c38ce-6599-4ff3-ae82-915b91a07db9",

                    "resultMedicalCare": "5",

                    "resultAmbulanceDepartureType": "3"

                },

                "currentStageId": "d41a2f57-6c06-4f38-9982-ea1fc7730d01",

                "currentStage": "Черновик",

                "workflowId": "b3393395-06f7-4261-b403-5dc0dab1296b",

                "workflowName": "Активы\_бп",

                "processName": "Новая заявка актив",

                "created": "2020-12-18T11:17:12.202303+03:00",

                "updated": "2020-12-18T11:17:12.202303+03:00",

                "businessStatus": {

                    "system": "urn:oid:1.2.643.2.69.1.1.1.148.2",

                    "code": "2"

                }

            },

            {

                "processId": "4bf133a0-549f-4e28-b0e1-3a44c9b0bbde",

                "processHumanFriendlyId": "TMC122052YL1B",

                "metadata": {

                    "patient": "8ff30a0b-85c3-462c-aae1-3ec719b3c1a3",

                    "performer": "fc2c38ce-6599-4ff3-ae82-915b91a07db9",

                    "resultMedicalCare": "3",

                    "resultAmbulanceDepartureType": "3"

                },

                "currentStageId": "d41a2f57-6c06-4f38-9982-ea1fc7730d01",

                "currentStage": "Черновик",

                "workflowId": "b3393395-06f7-4261-b403-5dc0dab1296b",

                "workflowName": "Активы\_бп",

                "processName": "Новая заявка актив",

                "created": "2020-12-18T11:58:30.200089+03:00",

                "updated": "2020-12-18T11:58:30.200089+03:00",

                "businessStatus": {

                    "system": "urn:oid:1.2.643.2.69.1.1.1.148.2",

                    "code": "2"

                }

            },

            {

                "processId": "727523ee-059e-40bf-a396-abcff9160287",

                "processHumanFriendlyId": "TMC122055BU7D",

                "metadata": {

                    "patient": "8ff30a0b-85c3-462c-aae1-3ec719b3c1a3",

                    "performer": "fc2c38ce-6599-4ff3-ae82-915b91a07db9",

                    "requester": "931a9317-586c-4dd5-bc32-cd8d3af78903",

                    "resultMedicalCare": "3",

                    "resultAmbulanceDepartureType": "3"

                },

                "currentStageId": "d41a2f57-6c06-4f38-9982-ea1fc7730d01",

                "currentStage": "Черновик",

                "workflowId": "b3393395-06f7-4261-b403-5dc0dab1296b",

                "workflowName": "Активы\_бп",

                "processName": "Новая заявка актив",

                "created": "2020-12-18T12:23:30.272053+03:00",

                "updated": "2020-12-18T12:23:30.272053+03:00",

                "businessStatus": {

                    "system": "urn:oid:1.2.643.2.69.1.1.1.148.2",

                    "code": "2"

                }

            }

        ],

        "total": 6

    },

    "success": true,

    "errorCode": 0,

    "message": null,

    "stackTrace": null

}

## Получение объекта контекста заявки (POST //GetProcessContext)

Для получения объекта контекста заявки в сервисе используется метод POST GetProcessContext. Метод предназначен для получения набора данных (объекта контекста), собранного при работе с заявкой.

### **Описание параметров запроса**

В таблице ниже представлено описание параметров запроса метода.

Таблица 20 Входные параметры для метода POST GetProcessContext

| **№ 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 () |

### Пример запроса

POST {{url}}/api/Queries/GetProcessContext

{

    "RoleContext": [

        {

            "Role": "DEPCHIEF",

            "Organization": "931a9317-586c-4dd5-bc32-cd8d3af78903"

        }

    ],

    "ProcessId": "64f88f61-7747-402c-9e04-d93eaadc234f"

}

## Получение маршрута обработки заявки по идентификатору (GetWorkflow)

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.

### **Описание параметров запроса и ответа**

В таблице ниже представлено описание параметров запроса метода.

Таблица 21 Входные параметры метода GET GetWorkflow

| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| --- | --- | --- | --- | --- |
| 1 | workflowId | String | 1..1 | Идентификатор маршрута |

Описание структуры ответа

Таблица 22 Выходные параметры метода GET GetWorkflow

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ 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 запроса

GET {{url}}/api/Queries/GetWorkflow/[id маршрута]

POST http://[hostname]/api/Queries/GetWorkflow/b3393395-06f7-4261-b403-5dc0dab1296b authorization: N3[пробел][GUID передающей системы]

content-type: application/json

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

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

### **Описание параметров запроса и ответа**

В таблице ниже представлено описание параметров запроса метода.

Таблица 23 Входные параметры для метода POST GetSchema

| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| --- | --- | --- | --- | --- |
| 1 | id | String | 1..1 | Идентификатор схемы |

Описание структуры ответа

В описании структур, передаваемых в ответе данных в формате json-схемы, содержится описания типов передаваемых данных, кратности и обязательности. 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, 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: Индекс пациента, ИЭМК, ОДЛИ, ОДИИ, УО.

### Пример запроса

GET {{url}}/api/Commands/GetSchema/[id схемы]

GET http://[hostname]/api/Queries/GetSchema/e57a49c0-717c-4d25-a045-114157567088

authorization: N3[пробел][GUID передающей системы]

content-type: application/json

## Организация обмена файлами. Компонент XDS

Компонент XDS предназначен для распределенного хранения данных, поступающих в функциональные сервисы платформы N3. Компонент представляет собой REST-сервис и представляет собой сайт IIS. Компонент распространяется как самостоятельный компонент в виде nuget-пакета вида N3.XDS.0.1.0-unstable0030.nupkg.

В текущем решении реализованы следующие методы:

* Регистрация документа в XDS.
* Получение документа из XDS.

### **Передача файла вложения заявки (POST //api/Commands/xds)**

Для передачи объекта файла вложения, прикрепленного к заявке (регистрация данных) в сервисе XDS используется метод POST {{url}}/api/Commands/xds, позволяющий загрузить файл и в результате получить ссылку (идентификатор) загруженного в ответ. Метод предназначен для отправки файлов, прикрепленных к заявке (например, результатов исследований, выполненных на вызове).

### **Описание параметров запроса**

В таблице ниже представлено описание параметров запроса метода.

Таблица 24 Входные параметры для метода POST //api/Commands/xds

| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| --- | --- | --- | --- | --- |
| 1 | formFile | string | 0..1 | Ссылка на файл, указывается для передачи расположение файла |

### Пример запроса

POST {{url}}/api/Commands/xds

{

formFile

}

### **Получение файла вложения заявки (POST //xds)**

Для получения объекта файла вложения, прикрепленного к заявке в сервисе, используется метод POST {{url}}/api/Queries/xds/{fileId}. Метод предназначен для получения файлов, прикрепленных к заявке (например, результатов исследований, выполненных на вызове), при выполнении следующих условий:

1. На входе передается идентификатор заявки, по которой необходимо получить файл, ссылку (идентификатор) на файл, ролевой контекст.
2. Система проверяет ролевой контекст для текущего состояния заявки, наличие идентификатора файла в контексте заявки.
3. Если оба условия удовлетворены, то пользователю предоставляется возможность получить файл.

### Описание параметров запроса

В таблице ниже представлено описание параметров запроса метода.

Таблица 25 Входные параметры для метода POST //xds

| **№ п/п** | **Параметр** | **Тип** | **Кратность** | **Описание** |
| --- | --- | --- | --- | --- |
| 1 | fileId | string | 0..1 | Идентификатор файла, полученный в ответе при отправке. Идентификатор указывается в контексте заявки. |
| 2 | RoleContext | array | 0..1 | Ролевой контекст для текущего состояния заявки в процессе |

### Пример запроса

POST {{url}}/api/Queries/xds/{fileId}

{

"RoleContext": [

{

"Role": "DEPCHIEF",

"Organization": "931a9317-586c-4dd5-bc32-cd8d3af78903"

}

]

"ProcessId": "b4fa714e-5d81-485a-bb22-16ae2b4ffe2f"

}

## Организация поддержки FHIR-стандарта в методах сервиса

Сервис предоставляет возможность обмена данными по стандарту FHIR. С целью упрощения процесса интеграции с сервисом Активы по стандарту FHIR реализовано два метода, позволяющие преобразовывать данные простого формата json к формату FHIR. Методы могут помочь при отладке в процессе интеграции. Использование методов преобразования не обязательно. Приведение к формату FHIR, если его применение предполагает обмен, можно выполнять самостоятельно.

Для конвертации данных в формат стандарта FHIR и обратно предназначены методы:

* /api/debug/convertSimpleJsonToFhirJson - преобразование из простой структуры в FHIR;
* /api/debug/convertFhirJsonToSimpleJson - обратное преобразование из FHIR в простую структуру.

Данный стандарт поддерживается для следующих методов обмена данными заявки /api/Fhir/StartNewProcess, /api/Fhir/MoveToStage, /api/Fhir/ProcessContext, /api/Fhir/Process/{processId}.

### **Общий принцип преобразования**

При организации обмена по стандарту FHIR для указанных выше методов работы с заявкой происходит преобразование данных processContext и roleContext к формату ресурсов FHIR questionnaireresponse и parameters соответственно.

Для преобразования в формат FHIR в адресе запроса необходимо указать тип ресурса в формат которого конвертируются данные:

/api/debug/convertSimpleJsonToFhirJson?fhirType=QuestionnaireResponse

или

/api/debug/convertSimpleJsonToFhirJson?fhirType=Parameters.

При этом в теле запроса передаются данные processContext и roleContext в стандартном формате json.

Сценарий преобразования страндартного json в формат FHIR предполагает следующие шаги:

1. Вызов метода преобразования контекста заявки processContext (/api/debug/convertSimpleJsonToFhirJson?fhirType=QuestionnaireResponse) и сохранение результата.
2. Вызов метода преобразования ролевого контекста roleContext (/api/debug/convertSimpleJsonToFhirJson?fhirType=Parameters) и сохранение результата.
3. Вызов метода обработки заявки FHIR (например, создания заявки POST /api/Fhir/StartNewProcess) и добавление в тело запроса результатов шагов 1 и 2.
4. Получение результата создания заявки по стандарту FHIR.

Примеры полного преобразования тела запроса по созданию заявки в FHIR в пункте 5.12.7.

Структуры данных заявки, ролевого контекста и метаданных приводятся к максимально утилитарному минималистичному формату json. Интерфейс на FHIR генерируется и интерпретируются автоматически. Данные передаваемые клиентскими системами при создании и изменении заявки интерпретируются как формы(анкеты), заполненные пользователем согласно формату ресурса questionnaireresponse стандарта FHIR и параметры пользователя для определения ролевого доступа – формат ресурса parameters стандарта FHIR.

Далее конвертированные результаты для processContext и roleContext можно использовать с методами обработки заявки по FHIR (См. описание в пунктах 5.12.4 - 5.12.6).

Для обратного преобразования предназначен метод /api/debug/convertFhirJsonToSimpleJson, в теле запроса которого передаются данные processContext и roleContext в формате FHIR. В результате обработки будут получены объекты processContext и roleContext приведенные к простому виду json, которые можно применять со стандартными методами работы с заявкой.

### **Обработка данных из processContext в FHIR**

При конвертации входящих данных из processContext к виду FHIR для методов startNewProcess и moveToStage преобразовываются структуры объекта "resourse", входящего в параметр запроса с именем "processContext" в структуры, которые хранятся физически.

Структура преобразовывается соответственно типам данных ресурса FHIR questionnaireresponse (подробное описание состава ресурса по ссылке <http://fhir-ru.github.io/questionnaireresponse.html>).

* "valueString": "<string>" – строки;
* "valueBoolean": <boolean> – булевые;
* "valueInteger": <integer> – числа;
* "item": [{}] - вложенные сложные структуры, массивы, объекты.

Для разделения и преобразования объектов и массивов внутри processContext применяется следующая логика:

* Узел типа "object" и все элементы, которые содержит объект при конвертации приводятся к виду:

{

"linkId": "object",

"answer": [

{

"item": []

}

]

}

Объект FHIR "answer" - может содержать внутри элементы или item или value (), где item - object, value - примитивный тип (string, bool, interger, ...). Если указаны оба варианты, то выводится ошибка. Также, если в объекте FHIR item лежит другой item, это обозначает, что вложенный объект item – массив, все элементы внутри которого будут иметь вид, принятый для данного типа элементов.

* Узел типа "array" и все элементы, которые содержит объект при конвертации приводятся к виду:

{

"linkId": "array",

"item": []

}

Массив может состоять как из примитивных типов, так и из объектов, все элементы внутри которых будут иметь вид, принятый для данного типа элементов.

Например, узел «объект»:

"patient": {

"patientIdMPI": "019802c4-7e56-4f60-a10b-34ba4b256219",

"patientName": "Гусев Евгений Викторович",

"patientemail": "89128329zxc813@mail.com"

}

Преобразовывается к виду:

{

"linkId": "patient",

"text": "object",

"item": [

{

"linkId": "patientIdMPI",

"answer": [

{

"valueString": "019802c4-7e56-4f60-a10b-34ba4b256219"

}

]

},

{

"linkId": "patientName",

"answer": [

{

"valueString": "Гусев Евгений Викторович"

}

]

},

{

"linkId": "patientemail",

"answer": [

{

"valueString": "89128329zxc813@mail.com"

}

]

}

]

}

Например, для узла «массив»:

"files": [

{

"id": "1",

"URL": "ya.ru",

"isDeleted": false

}

]

Преобразовывается к виду:

{

"linkId": "files",

"item": [

{

"linkId": "0",

"answer": [

{

"item": [

{

"linkId": "id",

"answer": [

{

"valueString": "1"

}

]

},

{

"linkId": "URL",

"answer": [

{

"valueString": "ya.ru"

}

]

},

{

"linkId": "isDeleted",

"answer": [

{

"valueBoolean": false

}

]

}

]

}

]

}

]

}

Элементы массива нумеруются по порядку "linkId": "0", так как узел "linkId" в FHIR является обязательным.

В методах /api/Fhir/StartNewProcess и /api/Fhir/MoveToStage сервис поддерживает и обратное преобразование данных из переданных в по стандарту FHIR к внутреннему формату.

При работе с методом /api/Fhir/ProcessContext так же поддерживается преобразование в обратную сторону из структуры, хранящейся в системе к формату FHIR.

Пример стандартной структуры TMCore:

{

"patient": {

"idMPI": "8ff30a0b-85c3-462c-aae1-3ec719b3c1a3",

"identityDocument": [

{

"id": "1",

"system": "1.2.643.2.69.1.1.1.6.14",

"code": "0003123123"

},

{

"id": "2",

"system": "1.2.643.2.69.1.1.1.6.228",

"code": "7849500830000203"

}

],

"birthDate": "2002-11-14",

"gender": "male",

"social": {

"registration": "1",

"socialStatus": "1"

},

"placeOfWork": "Google",

"address": {

"district": "район",

"city": "город/село",

"line": "ул. Наименование улицы,д.№"

},

"fullName": "Вакуленко Борис Владимирович",

"contact": {

"name": "ФИО вызывавшего",

"telecom": "125-25-25"

}

}

}

Пример структуры FHIR преобразованной по структуре выше:

{

"resourceType": "QuestionnaireResponse",

"status": "completed",

"item": [

{

"linkId": "patient",

"answer": [

{

"item": [

{

"linkId": "idMPI",

"answer": [

{

"valueString": "8ff30a0b-85c3-462c-aae1-3ec719b3c1a3"

}

]

},

{

"linkId": "identityDocument",

"item": [

{

"linkId": "0",

"answer": [

{

"item": [

{

"linkId": "id",

"answer": [

{

"valueString": "1"

}

]

},

{

"linkId": "system",

"answer": [

{

"valueString": "1.2.643.2.69.1.1.1.6.14"

}

]

},

{

"linkId": "code",

"answer": [

{

"valueString": "0003123123"

}

]

}

]

}

]

},

{

"linkId": "1",

"answer": [

{

"item": [

{

"linkId": "id",

"answer": [

{

"valueString": "2"

}

]

},

{

"linkId": "system",

"answer": [

{

"valueString": "1.2.643.2.69.1.1.1.6.228"

}

]

},

{

"linkId": "code",

"answer": [

{

"valueString": "7849500830000203"

}

]

}

]

}

]

}

]

},

{

"linkId": "birthDate",

"answer": [

{

"valueString": "2002-11-14"

}

]

},

{

"linkId": "gender",

"answer": [

{

"valueString": "male"

}

]

},

{

"linkId": "social",

"answer": [

{

"item": [

{

"linkId": "registration",

"answer": [

{

"valueString": "1"

}

]

},

{

"linkId": "socialStatus",

"answer": [

{

"valueString": "1"

}

]

}

]

}

]

},

{

"linkId": "placeOfWork",

"answer": [

{

"valueString": "Google"

}

]

},

{

"linkId": "address",

"answer": [

{

"item": [

{

"linkId": "district",

"answer": [

{

"valueString": "район"

}

]

},

{

"linkId": "city",

"answer": [

{

"valueString": "город/село"

}

]

},

{

"linkId": "line",

"answer": [

{

"valueString": "ул. Наименование улицы,д.№"

}

]

}

]

}

]

},

{

"linkId": "fullName",

"answer": [

{

"valueString": "Вакуленко Борис Владимирович"

}

]

},

{

"linkId": "contact",

"answer": [

{

"item": [

{

"linkId": "name",

"answer": [

{

"valueString": "ФИО вызывавшего"

}

]

},

{

"linkId": "telecom",

"answer": [

{

"valueString": "125-25-25"

}

]

}

]

}

]

}

]

}

]

}

]

}

### **Обработка данных из roleContext в FHIR**

При конвертации входящих данных из roleContext к виду FHIR для методов startNewProcess и moveToStage преобразовываются структуры объекта ролевого контекста в структуры, которые хранятся физически.

Структура преобразовываются в соответствии типов данных ресурса FHIR parameters (подробное описание состава ресурса по ссылке <http://fhir-ru.github.io/parameters.html>).

Для разделения и преобразования объектов и массивов внутри roleContext применяется следующая логика:

* Узел типа "object" и все элементы, которые содержит объект при конвертации приводятся к виду:

{

"name": "object",

"resource": {

"resourceType": "Parameters",

"parameter": [

{

"name": "element",

"valueString": "value"

}

]

}

}

Объект FHIR "parameter" - может содержать внутри и элементы, и значения элементов value (), где value - примитивный тип (string, bool, interger, ...).

* Узел типа "array" и все элементы, которые содержит объект при конвертации приводятся к виду:

{

"name": "array",

"part": [

{

"name": "0",

"resource": {

"resourceType": "Parameters",

"parameter": [

{

"name": "Role",

"valueString": "DOCTOR"

}

]

}

]

}

Объект FHIR "part" - может содержать внутри объекты, все элементы внутри которых будут иметь вид, принятый для данного типа элементов.

Пример структуры ролевого контекста TMCore:

{

    "RoleContext": [

        {

            "Role": "DEPCHIEF",

            "Organization": "931a9317-586c-4dd5-bc32-cd8d3af78903"

        }

    ]

}

Пример структуры ролевого контекста по FHIR преобразованной из структуры выше:

{

    "resourceType": "Parameters",

    "parameter": [

        {

            "name": "RoleContext",

            "part": [

                {

                    "name": "0",

                    "resource": {

                        "resourceType": "Parameters",

                        "parameter": [

                            {

                                "name": "Role",

                                "valueString": "DEPCHIEF"

                            },

                            {

                                "name": "Organization",

                                "valueString": "931a9317-586c-4dd5-bc32-cd8d3af78903"

                            }

                        ]

                    }

                }

            ]

        }

    ]

}

### **Описание метода создания заявки по стандарту FHIR (POST //api/Fhir/StartNewProcess)**

Метод предназначен для создания заявки об активном вызове по стандарту FHIR. В таблице ниже представлено описание параметров запроса метода.

Таблица 26 Входные параметры для метода POST /api/Fhir/StartNewProcess

| **№ 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 (структура, смотреть в контрольных примерах). Может быть получен в результате преобразования данных к формату FHIR |
| 5 | roleContext | Object | 1..1 | Значение ролевого контекста пользователя. Может быть получен в результате преобразования данных к формату FHIR |

### Пример формата запроса по стандарту Fhir

{

"resourceType": "Parameters",

"parameter": [

{

"name": "workflowId",

"valueString": "00000000-0000-0000-0000-000000000000",

},

{

"name": "name",

"valueString": null,

},

{

"name": "initialTransitionId",

"valueString": null,

},

{

"name": "processContext",

"resource": {}

},

{

"name": "roleContext",

"part": "[{}]"

}

]

}

Полный пример запроса представлен в приложениии - **Ошибка! Источник ссылки не найден.**.

### **Описание метода редактирования/обновления заявки по стандарту FHIR (POST //api/Fhir/MoveToStage)**

Метод предназначен для редактирования и обновления заявки об активном вызове по маршруту по стандарту Fhir. В таблице ниже представлено описание параметров запроса метода.

Таблица 27 Входные параметры для метода POST /api/Fhir/MoveToStage

| **№ 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 (структура, смотреть в контрольных примерах). Может быть получен в результате преобразования данных к формату FHIR |
| 4 | roleContext | Object | 1..1 | Значение ролевого контекста пользователя. Может быть получен в результате преобразования данных к формату FHIR |

### Пример формата запроса по стандарту Fhir

{

"resourceType": "Parameters",

"parameter": [

{

"name": "processId",

"valueString": "00000000-0000-0000-0000-000000000000",

},

{

"name": "transitionId",

"valueString": "00000000-0000-0000-0000-000000000000",

},

{

"name": "processContext",

"resource": {}

},

{

"name": "roleContext",

"part": "[{}]",

}

]

}

Полный пример запроса представлен в приложениии - **Ошибка! Источник ссылки не найден.**.

### **Описание метода получения объекта контекста заявки по стандарту FHIR (POST //api/Fhir/ProcessContext)**

Метод предназначен для получения объекта контекста заявки по стандарту Fhir. В таблице ниже представлено описание параметров запроса метода.

Таблица 28 Входные параметры для метода POST //api/Fhir/ProcessContext

| **№ 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 |

### Пример формата запроса по стандарту Fhir

{

"resourceType": "Parameters",

"parameter": [

{

"name": "processId",

"valueString": "00000000-0000-0000-0000-000000000000",

},

{

"name": "roleContext",

"part": "[{}]",

}

]}

Полный пример запроса представлен в приложениии - **Ошибка! Источник ссылки не найден.**.

### Пример полного преобразования тела запроса по созданию заявки в FHIR

{

"resourceType": "Parameters",

"parameter": [

{

"name": "workflowId",

"valueString": "b3393395-06f7-4261-b403-5dc0dab1296b"

},

{

"name": "name",

"valueUrl": "Sz-process6-Fhir"

},

{

"name": "initialTransitionId",

"valueUrl": "e9b2cce6-93ec-4118-8728-1580af7b9e82"

},

{

"name": "processContext",

"resource": {

"resourceType": "QuestionnaireResponse",

"status": "completed",

"item": [

{

"linkId": "patient",

"answer": [

{

"item": [

{

"linkId": "idMPI",

"answer": [

{

"valueString": "8ff30a0b-85c3-462c-aae1-3ec719b3c1a3"

}

]

},

{

"linkId": "identityDocument",

"item": [

{

"linkId": "0",

"answer": [

{

"item": [

{

"linkId": "id",

"answer": [

{

"valueString": "1"

}

]

},

{

"linkId": "system",

"answer": [

{

"valueString": "1.2.643.2.69.1.1.1.6.14"

}

]

},

{

"linkId": "code",

"answer": [

{

"valueString": "0003123123"

}

]

}

]

}

]

},

{

"linkId": "1",

"answer": [

{

"item": [

{

"linkId": "id",

"answer": [

{

"valueString": "2"

}

]

},

{

"linkId": "system",

"answer": [

{

"valueString": "1.2.643.2.69.1.1.1.6.228"

}

]

},

{

"linkId": "code",

"answer": [

{

"valueString": "7849500830000203"

}

]

}

]

}

]

}

]

},

{

"linkId": "birthDate",

"answer": [

{

"valueString": "2002-11-14"

}

]

},

{

"linkId": "gender",

"answer": [

{

"valueString": "male"

}

]

},

{

"linkId": "social",

"answer": [

{

"item": [

{

"linkId": "registration",

"answer": [

{

"valueString": "1"

}

]

},

{

"linkId": "socialStatus",

"answer": [

{

"valueString": "1"

}

]

}

]

}

]

},

{

"linkId": "placeOfWork",

"answer": [

{

"valueString": "Google"

}

]

},

{

"linkId": "address",

"answer": [

{

"item": [

{

"linkId": "district",

"answer": [

{

"valueString": "район"

}

]

},

{

"linkId": "city",

"answer": [

{

"valueString": "город/село"

}

]

},

{

"linkId": "line",

"answer": [

{

"valueString": "ул. Наименование улицы,д.№"

}

]

}

]

}

]

},

{

"linkId": "fullName",

"answer": [

{

"valueString": "Вакуленко Борис Владимирович"

}

]

},

{

"linkId": "contact",

"answer": [

{

"item": [

{

"linkId": "name",

"answer": [

{

"valueString": "ФИО вызывавшего"

}

]

},

{

"linkId": "telecom",

"answer": [

{

"valueString": "125-25-25"

}

]

}

]

}

]

}

]

}

]

},

{

"linkId": "dispatcherRole",

"answer": [

{

"item": [

{

"linkId": "identityDocument",

"answer": [

{

"item": [

{

"linkId": "system",

"answer": [

{

"valueString": "1.2.643.2.69.1.1.1.6.223"

}

]

},

{

"linkId": "code",

"answer": [

{

"valueString": "48835311311"

}

]

}

]

}

]

},

{

"linkId": "dispatcherNumber",

"answer": [

{

"valueString": "АБ-23"

}

]

},

{

"linkId": "organization",

"answer": [

{

"valueString": "931a9317-586c-4dd5-bc32-cd8d3af78903"

}

]

}

]

}

]

},

{

"linkId": "paramedicRole",

"answer": [

{

"item": [

{

"linkId": "identityDocument",

"answer": [

{

"item": [

{

"linkId": "system",

"answer": [

{

"valueString": "1.2.643.2.69.1.1.1.6.223"

}

]

},

{

"linkId": "code",

"answer": [

{

"valueString": "48835311311"

}

]

}

]

}

]

},

{

"linkId": "organization",

"answer": [

{

"valueString": "931a9317-586c-4dd5-bc32-cd8d3af78903"

}

]

}

]

}

]

},

{

"linkId": "seniorParamedicRole",

"answer": [

{

"item": [

{

"linkId": "identityDocument",

"answer": [

{

"item": [

{

"linkId": "system",

"answer": [

{

"valueString": "1.2.643.2.69.1.1.1.6.223"

}

]

},

{

"linkId": "code",

"answer": [

{

"valueString": "48835311311"

}

]

}

]

}

]

},

{

"linkId": "organization",

"answer": [

{

"valueString": "931a9317-586c-4dd5-bc32-cd8d3af78903"

}

]

}

]

}

]

},

{

"linkId": "depChiefRole",

"answer": [

{

"item": [

{

"linkId": "identityDocument",

"answer": [

{

"item": [

{

"linkId": "system",

"answer": [

{

"valueString": "1.2.643.2.69.1.1.1.6.223"

}

]

},

{

"linkId": "code",

"answer": [

{

"valueString": "48835311311"

}

]

}

]

}

]

},

{

"linkId": "organization",

"answer": [

{

"valueString": "931a9317-586c-4dd5-bc32-cd8d3af78903"

}

]

}

]

}

]

},

{

"linkId": "serviceRequest",

"answer": [

{

"item": [

{

"linkId": "reasonCallMedicalCare",

"answer": [

{

"valueString": "2"

}

]

},

{

"linkId": "territoryCode",

"answer": [

{

"valueString": "3"

}

]

},

{

"linkId": "requesterOrganization",

"answer": [

{

"valueString": "931a9317-586c-4dd5-bc32-cd8d3af78903"

}

]

},

{

"linkId": "brigadeNumber",

"answer": [

{

"valueString": "9317"

}

]

},

{

"linkId": "shiftNumber",

"answer": [

{

"valueString": "1"

}

]

},

{

"linkId": "placeCallReceived",

"answer": [

{

"valueString": "1"

}

]

},

{

"linkId": "reasonsBeingLate",

"answer": [

{

"valueString": "2"

}

]

},

{

"linkId": "ambulanceTeam",

"answer": [

{

"valueString": "3"

}

]

},

{

"linkId": "accidentCause",

"answer": [

{

"valueString": "3"

}

]

},

{

"linkId": "intoxication",

"answer": [

{

"valueBoolean": false

}

]

},

{

"linkId": "complaint",

"answer": [

{

"valueString": "жалобы"

}

]

},

{

"linkId": "complicationsExamination",

"answer": [

{

"item": [

{

"linkId": "complications",

"answer": [

{

"valueString": "2"

}

]

},

{

"linkId": "complicationsCodMKB",

"answer": [

{

"valueString": "T78.2"

}

]

},

{

"linkId": "effectivenessMeasures",

"answer": [

{

"valueString": "2"

}

]

},

{

"linkId": "assistanceProvidedLocationCallАmbulance",

"answer": [

{

"valueString": "оказанная помощь на месте вызова"

}

]

},

{

"linkId": "assistanceProvidedАmbulanceCar",

"answer": [

{

"valueString": "оказанная помощь в автомобиле скорой"

}

]

},

{

"linkId": "performanceIndicatorsEvents",

"answer": [

{

"item": [

{

"linkId": "bloodPressure",

"answer": [

{

"valueString": "120/80"

}

]

},

{

"linkId": "pulse",

"answer": [

{

"valueInteger": 60

}

]

},

{

"linkId": "heartRate",

"answer": [

{

"valueInteger": 30

}

]

},

{

"linkId": "respirationRate",

"answer": [

{

"valueInteger": 40

}

]

},

{

"linkId": "bodyHeat",

"answer": [

{

"valueString": "36.6"

}

]

},

{

"linkId": "pulseOximetry",

"answer": [

{

"valueString": "23.3"

}

]

},

{

"linkId": "glucose monitoring",

"answer": [

{

"item": [

{

"linkId": "bloodGlucoseLowerBound",

"answer": [

{

"valueString": "100.54"

}

]

},

{

"linkId": "bloodGlucoseUpperBound",

"answer": [

{

"valueString": "12.34"

}

]

}

]

}

]

}

]

}

]

}

]

}

]

},

{

"linkId": "deliveryMethodToCar",

"answer": [

{

"valueString": "1"

}

]

},

{

"linkId": "resultAmbulanceDepartureType",

"answer": [

{

"valueString": "3"

}

]

},

{

"linkId": "kilometers",

"answer": [

{

"valueString": "километраж"

}

]

},

{

"linkId": "time",

"answer": [

{

"item": [

{

"linkId": "receiveСallDateTime",

"answer": [

{

"valueDateTime": "2010-10-07T13:41:23+04:00"

}

]

},

{

"linkId": "callЕransferDateTime",

"answer": [

{

"valueDateTime": "2010-10-08T13:43:23+04:00"

}

]

},

{

"linkId": "checkoutTime",

"answer": [

{

"valueString": "hh:mm:ss"

}

]

},

{

"linkId": "arrivalTime",

"answer": [

{

"valueString": "hh:mm:ss"

}

]

},

{

"linkId": "startTimeTransportation",

"answer": [

{

"valueString": "hh:mm:ss"

}

]

},

{

"linkId": "arrivalTimeСlinic",

"answer": [

{

"valueString": "hh:mm:ss"

}

]

},

{

"linkId": "endTimeCall",

"answer": [

{

"valueString": "hh:mm:ss"

}

]

},

{

"linkId": "timeReturnToStation",

"answer": [

{

"valueString": "hh:mm:ss"

}

]

},

{

"linkId": "timeTakenComplete",

"answer": [

{

"valueString": "затраченное на вызов время"

}

]

}

]

}

]

},

{

"linkId": "ambulanceCallType",

"answer": [

{

"valueString": "1"

}

]

},

{

"linkId": "resultMedicalCare",

"answer": [

{

"valueString": "5"

}

]

},

{

"linkId": "performerOrganization",

"answer": [

{

"valueString": "fc2c38ce-6599-4ff3-ae82-915b91a07db9"

}

]

},

{

"linkId": "hospitalOrganization",

"answer": [

{

"valueString": "fc2c38ce-6599-4ff3-ae82-915b91a07db9"

}

]

},

{

"linkId": "idIEMK",

"answer": [

{

"valueString": "1110025"

}

]

},

{

"linkId": "consent",

"answer": [

{

"valueBoolean": true

}

]

},

{

"linkId": "comments",

"answer": [

{

"valueString": "доп.текст.инфа"

}

]

}

]

}

]

},

{

"linkId": "observation",

"answer": [

{

"item": [

{

"linkId": "generalCondition",

"answer": [

{

"valueString": "1"

}

]

},

{

"linkId": "behavior",

"answer": [

{

"valueString": "2"

}

]

},

{

"linkId": "consciousness",

"answer": [

{

"valueString": "2"

}

]

},

{

"linkId": "meningealSigns",

"answer": [

{

"valueString": "5"

}

]

},

{

"linkId": "eyes",

"answer": [

{

"item": [

{

"linkId": "pupils",

"answer": [

{

"valueString": "3"

}

]

},

{

"linkId": "anisocoria",

"answer": [

{

"valueString": "4"

}

]

},

{

"linkId": "nystagmus",

"answer": [

{

"valueString": "5"

}

]

},

{

"linkId": "lightResponse",

"answer": [

{

"valueString": "6"

}

]

}

]

}

]

},

{

"linkId": "skin",

"answer": [

{

"item": [

{

"linkId": "integument",

"answer": [

{

"valueString": "2"

}

]

},

{

"linkId": "acrocyanosis",

"answer": [

{

"valueString": "6"

}

]

},

{

"linkId": "marbling",

"answer": [

{

"valueString": "9"

}

]

},

{

"linkId": "edemas",

"answer": [

{

"valueString": "10"

}

]

},

{

"linkId": "localizationOfEdema",

"answer": [

{

"valueString": ""

}

]

},

{

"linkId": "rash",

"answer": [

{

"valueString": "13"

}

]

},

{

"linkId": "rashLocalization",

"answer": [

{

"valueString": ""

}

]

}

]

}

]

},

{

"linkId": "breathing",

"answer": [

{

"valueString": "2"

}

]

},

{

"linkId": "wheezing",

"answer": [

{

"valueString": "2"

}

]

},

{

"linkId": "shortnessOfBreath",

"answer": [

{

"valueString": "3"

}

]

},

{

"linkId": "organsCirculatorySystem",

"answer": [

{

"item": [

{

"linkId": "heartSounds",

"answer": [

{

"valueString": "2"

}

]

},

{

"linkId": "noise",

"answer": [

{

"valueString": "1"

}

]

},

{

"linkId": "pulseLike",

"answer": [

{

"valueString": "1"

}

]

}

]

}

]

},

{

"linkId": "digestiveOrgans",

"answer": [

{

"item": [

{

"linkId": "tongue",

"answer": [

{

"valueString": "1"

}

]

},

{

"linkId": "belly",

"answer": [

{

"valueString": "1"

}

]

},

{

"linkId": "bellyInActOfBreathing",

"answer": [

{

"valueString": "7"

}

]

},

{

"linkId": "peritonealIrritationSymptoms",

"answer": [

{

"valueString": "8"

}

]

},

{

"linkId": "enlargedLiver",

"answer": [

{

"valueString": "11"

}

]

}

]

}

]

},

{

"linkId": "urination",

"answer": [

{

"valueString": ""

}

]

},

{

"linkId": "feces",

"answer": [

{

"valueString": ""

}

]

},

{

"linkId": "otherSymptoms",

"answer": [

{

"valueString": ""

}

]

},

{

"linkId": "pressure",

"answer": [

{

"item": [

{

"linkId": "workingBloodPressure",

"answer": [

{

"valueString": "120/80"

}

]

},

{

"linkId": "bloodPressure",

"answer": [

{

"valueString": "120/80"

}

]

}

]

}

]

},

{

"linkId": "pulse",

"answer": [

{

"valueInteger": 60

}

]

},

{

"linkId": "heartRate",

"answer": [

{

"valueInteger": 30

}

]

},

{

"linkId": "respirationRate",

"answer": [

{

"valueInteger": 40

}

]

},

{

"linkId": "bodyHeat",

"answer": [

{

"valueString": "36.6"

}

]

},

{

"linkId": "pulseOximetry",

"answer": [

{

"valueString": "23.3"

}

]

},

{

"linkId": "glucose monitoring",

"answer": [

{

"item": [

{

"linkId": "bloodGlucoseLowerBound",

"answer": [

{

"valueString": "100.54"

}

]

},

{

"linkId": "bloodGlucoseUpperBound",

"answer": [

{

"valueString": "12.34"

}

]

}

]

}

]

},

{

"linkId": "additionalObjectiveData",

"answer": [

{

"valueString": ""

}

]

},

{

"linkId": "ECG",

"answer": [

{

"item": [

{

"linkId": "preMedicalCareECG",

"answer": [

{

"valueString": ""

}

]

},

{

"linkId": "preMedicalCareECGTime",

"answer": [

{

"valueString": "01:30:00"

}

]

},

{

"linkId": "postMedicalCareECG",

"answer": [

{

"valueString": ""

}

]

},

{

"linkId": "postMedicalCareECGTime",

"answer": [

{

"valueString": "01:30:00"

}

]

}

]

}

]

}

]

}

]

},

{

"linkId": "condition",

"answer": [

{

"item": [

{

"linkId": "codeMKB",

"answer": [

{

"valueString": "S72.1"

}

]

},

{

"linkId": "complaints",

"answer": [

{

"valueString": "Жалобы"

}

]

},

{

"linkId": "anamnesis",

"answer": [

{

"valueString": "Дополнительные сведения по анамнезу заболевания"

}

]

}

]

}

]

},

{

"linkId": "attachedfiles",

"item": [

{

"linkId": "0",

"answer": [

{

"item": [

{

"linkId": "id",

"answer": [

{

"valueString": "1"

}

]

},

{

"linkId": "fileURL",

"answer": [

{

"valueString": ""

}

]

},

{

"linkId": "signatureURL",

"answer": [

{

"valueString": ""

}

]

},

{

"linkId": "isBlocked",

"answer": [

{

"valueBoolean": false

}

]

}

]

}

]

},

{

"linkId": "1",

"answer": [

{

"item": [

{

"linkId": "id",

"answer": [

{

"valueString": "2"

}

]

},

{

"linkId": "fileURL",

"answer": [

{

"valueString": ""

}

]

},

{

"linkId": "signatureURL",

"answer": [

{

"valueString": ""

}

]

},

{

"linkId": "isBlocked",

"answer": [

{

"valueBoolean": false

}

]

}

]

}

]

}

]

},

{

"linkId": "110/u",

"answer": [

{

"item": [

{

"linkId": "fileURL",

"answer": [

{

"valueString": ""

}

]

},

{

"linkId": "signatureURL",

"answer": [

{

"valueString": ""

}

]

}

]

}

]

}

]

}

},

{

"name": "RoleContext",

"part": [

{

"name": "0",

"resource": {

"resourceType": "Parameters",

"parameter": [

{

"name": "Role",

"valueInteger": 0

},

{

"name": "Organization",

"valueInteger": 0

}

]

}

}

]

}

]

}

# 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 |