

ОПИСАНИЕ ИНТЕГРАЦИОННЫХ ПРОФИЛЕЙ
СЕРВИС «ТЕЛЕМЕДИЦИНА»

ЛИСТОВ 42

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ.....	2
1. ОПИСАНИЕ РЕШЕНИЯ.....	5
1.1. Описание процесса обмена данными о заявках.....	5
1.2. Описание базового сценария обмена.....	5
2. ОПИСАНИЕ ПРОТОКОЛА ВЗАИМОДЕЙСТВИЯ	7
2.1. Внутренняя логика сервиса	7
2.1.1. Описание данных.....	7
2.1.2. Маршрут и его описание	7
2.2. Использование справочников.....	8
2.3. ПРЕДВАРИТЕЛЬНЫЕ УСЛОВИЯ	9
3. ПОЯСНЕНИЕ О РОЛЕВОЙ МОДЕЛИ.....	10
3.1. Доступ участнику информационного обмена (sign-in).....	10
3.2. Добавление и изменение постоянного пользователя.....	10
3.2.1. Добавление пользователя (registerUser).....	10
3.2.2. Чтение данных пользователя (GetUser).....	11
3.2.3. Изменение данных пользователя (updateUser)	12
3.3. Авторизация постоянных пользователей (sign-in)	13
3.4. Добавление пользователей-пациентов	13
3.4.1. Создание пациента (CreatePatient)	13
3.4.2. Получение данных о пациенте по идентификатору (SearchPatientById)	15
4. ОПИСАНИЕ МЕТОДОВ	20
4.1. Получение списка переходов доступных для создания заявки (GetAvailableTransitions).....	20
4.2. Получение схемы данных, передаваемых при осуществлении перехода (GetSchema).....	22
4.3. Создание новой заявки (StartNewProcess).....	24
4.4. Получение списка всех заявок, доступных пользователю для осуществления перехода (GetTransitionAvailableProcesses)	26
4.5. Получение контекста заявки (ProcessContext).....	29
4.6. Получение списка переходов доступных для заявки (ProcessWithAvailableTransitions)	30

4.7. Редактирование/обновление заявки (MoveToStage)	32
4.8. Получение списка доступных для просмотра заявок (GetReadAvailableProcesses)	34
4.9. Получение заявки по идентификатору (Process)	36
4.10. Организация обмена файлами. Компонент XDS	38
4.10.1. Передача файла вложения заявки (POST /api/Xds/xds)	38
4.10.2. Получение метаданных файла вложения заявки (GET /api/Xds/{id}/info)	39
4.10.3. Получение файла вложения (GET /api/Xds/xds).....	41
5. ОРГАНИЗАЦИЯ ПОЛУЧЕНИЯ ЧАТА ВКС	41
6. КОДЫ ВОЗВРАЩАЕМЫХ ОШИБОК.....	42

Определения, обозначения и сокращения

Термин	Содержание
МО	Медицинская организация
МИС МО	Медицинская информационная система медицинской организации
НСИ	«Сервис Терминологии» платформы N3.Health (сервис нормативно-справочной информации)
ТМ	Сервис «Телемедицина» платформы N3.Health
Индекс пациента (MPI)	Сервис «Индекс пациента» платформы N3.Health
СНИЛС	Страховой номер индивидуального лицевого счёта, СНИЛС — уникальный номер индивидуального лицевого счёта застрахованного лица в системе обязательного пенсионного страхования
МКБ-10	Международный классификатор болезней
ЭЦП	Электронная цифровая подпись

Версии документа

Версия	Дата	Описание
1.0	04.07.2024	Первоначальная версия документа
1.1	06.02.2025	Скорректировано описание компонента XDS, работы с чатом и ВКС.

1. ОПИСАНИЕ РЕШЕНИЯ

1.1. Описание процесса обмена данными о заявках

Сервис ТМ предназначен для обмена информацией по заявкам в рамках указанного процесса. Для разных типов заявок МО может выступать как в роли направляющей МО, так и в роли целевой (консультирующей) МО.

ТМ предоставляет возможность реализации различных сценариев работы с заявками. Сценарии зависят от набора операций, доступных для заявки в рамках указанного процесса. Под операциями понимаются различные действия над заявкой, такие как передача заявки консультирующему специалисту, возврат заявителю на дополнение, отказ в консультации по заявке, дополнение заявки сведениями, формирование заключения по заявке и т.д.

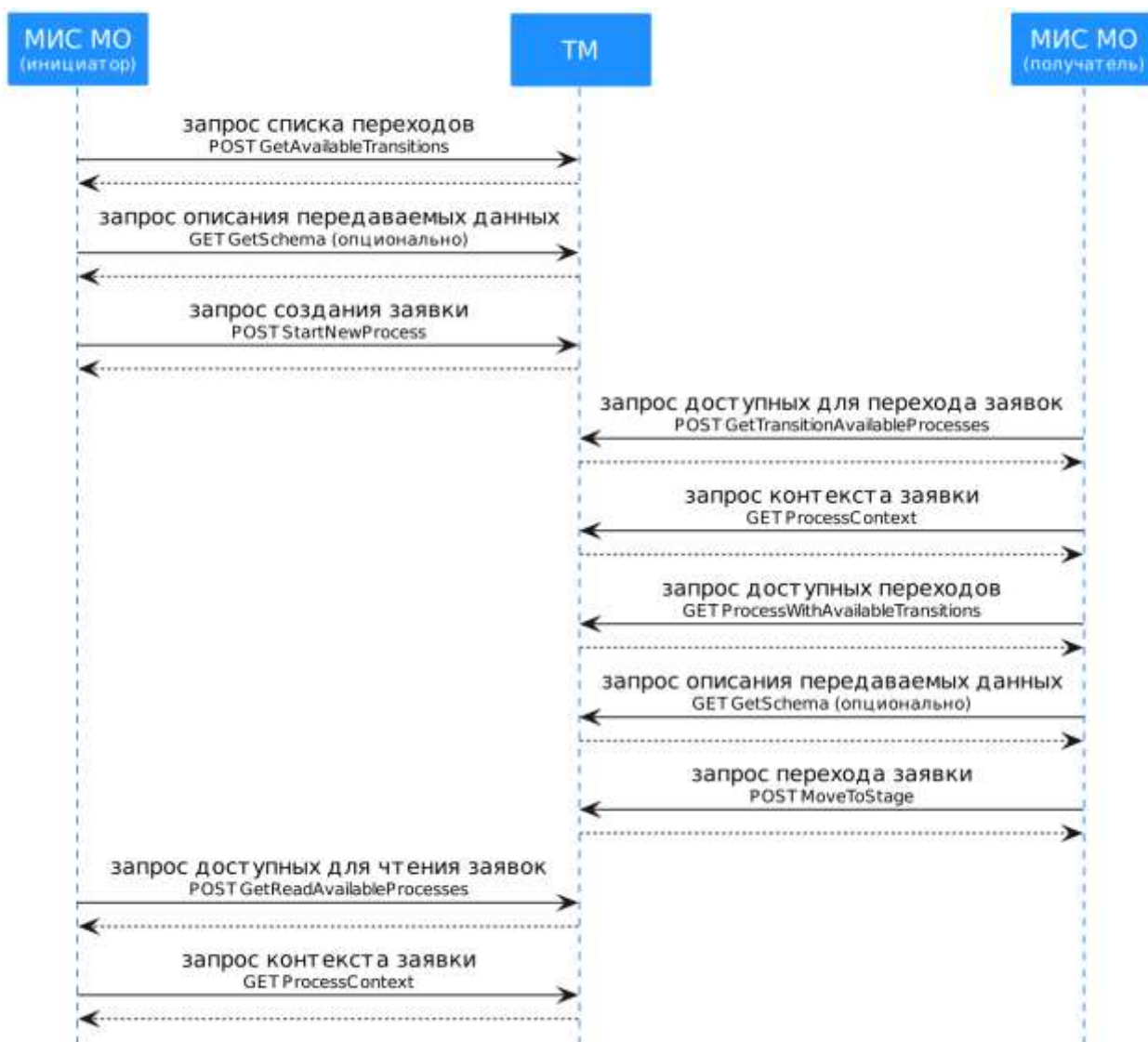
1.2. Описание базового сценария обмена

Сервис ТМ предназначен для получения, ведения, хранения, поиска и выдачи сведений о заявках согласно контексту бизнес-процесса.

Обмен данными между МИС МО и сервисом ТМ осуществляется в результате выполнения шагов следующего базового сценария:

1. Получить из сервиса список операций, доступных для создания и продвижения заявки.
2. По маршруту выбрать из сервиса описание условий перехода к новому состоянию заявки. На основе полученных данных выполнить запрос схем передачи данных, необходимых для осуществления операции: формат структуры передаваемых в заявку данных, используемых при описании характеристик пользователя.
3. Создать заявку по указанному маршруту.
4. Запросить список заявок, доступных для осуществления операций согласно их текущему состоянию. МИС МО запрашивает наличие заявок у сервиса ТМ для последующих действий.
5. Получить из сервиса контекст по заявке для принятия решения о последующем действии. МИС МО запрашивает объект контекста с содержанием заявки по ее идентификатору.
6. Выполнить запрос на изменение заявки для перевода её в следующее состояние в рамках указанного процесса. Выполнение производится до тех пор, пока заявка не будет приведена к конечному состоянию.
7. Запросить список заявок доступных для просмотра. МИС МО запрашивает список заявок в сервис ТМ для просмотра изменений по заявкам.
8. Получить из сервиса контекст по закрытой заявке для получения заключения.

Рисунок 1 – Описание базового сценария взаимодействия с сервисом «Телемедицина»



Детальное описание каждого маршрута, включая идентификаторы всех возможных переходов и структуру данных, передаваемых на этих переходах, представлено в документе «Общие сведения о маршруте» (спецификации бизнес-процесса обработки заявок), который предоставляется администратором сервиса.

2. ОПИСАНИЕ ПРОТОКОЛА ВЗАИМОДЕЙСТВИЯ

2.1. Внутренняя логика сервиса

2.1.1. Описание данных

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

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.

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.

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

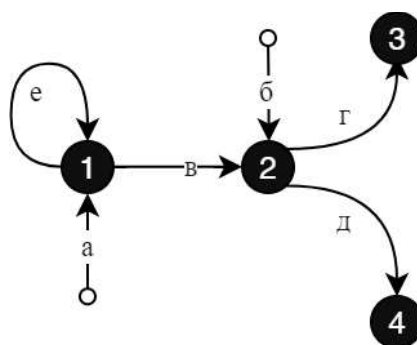
Данные, используемые при операциях с заявками – json-схема, описывающая набор и структуру данных необходимых для передачи со стороны пользователя при совершении операции с заявкой. Можно создать в рамках предметной области столько, сколько требуется.

Роли – json-схема, описывающая набор и структуру данных необходимых для передачи информации о пользователе при проверке доступа к функциям сервиса. Можно создать в рамках предметной области столько, сколько требуется. Пользователь может на входе передавать несколько наборов присвоенных ему ролей.

2.1.2. Маршрут и его описание

Маршрут – описание порядка выполнения операций в ходе исполнения работ по заявкам. Маршрут описывается с помощью названия, описания и структуры метаданных. Суть маршрута заключается в наборе состояний и операций с заявками.

Рисунок 2 – Схема состояний и операций над заявкой



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 соответствует конечному (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).

Каждое из состояний заявки имеет свой уникальный идентификатор в сервисе ТМ.

Содержание правил 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.

2.2. Использование справочников

Справочники, используемые в сервисе ТМ, размещены в сервисе НСИ. Подробное описание сервиса НСИ и правила его использования доступны по ссылке:

<https://api.n3health.ru/terminologyservice/>.

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": "<OID справочника>"
}
```

Медицинские организации передаются согласно справочнику: 1.2.643.2.69.1.1.1.64.

Полный перечень используемых сервисом справочников НСИ перечислен в описании параметров объектов.

2.3. ПРЕДВАРИТЕЛЬНЫЕ УСЛОВИЯ

Для взаимодействия с сервисом требуется получить у администраторов сервиса следующие данные:

1. Токен доступа участника информационного обмена (**token**)
2. Идентификатор пользователя - участника информационного обмена (**adminUserId**)
3. Спецификацию бизнес-процессов (маршрутов) обработки заявок, включающую в том числе идентификаторы ролевых схем (**roleSchemaId**)

3. ПОЯСНЕНИЕ О РОЛЕВОЙ МОДЕЛИ

Функционал сервиса расписаний предоставляется пользователям в соответствии с их ролевым контекстом, который опосредованно передается с помощью идентификатора сессии. Общий сценарий взаимодействия такой:

1. Постоянный пользователь (в том числе клиентская система) предварительно регистрируется в системе, ему присваивается ролевой контекст. Перед использованием сервиса, пользователь аутентифицируется в системе и получает временный идентификатор сессии, который передает в сервис.
2. Сервис восстанавливает по идентификатору сессии ролевой контекст пользователя и авторизует ему доступ к функциям и данным в соответствии с информации ролевого контекста.

Список доступных ролевых схем приводится в спецификации бизнес-процесса, которую необходимо предварительно получить у администраторов сервиса.

3.1. Доступ участнику информационного обмена (sign-in).

Для выполнения действий с пользователями требуется получить идентификатор сессии взаимодействия с сервисом ТМ, используя следующий метод:

```
POST BaseUrl/tm-widgets/api/sign-in
accept: text/plain
Content-Type: application/json-patch+json
Authorization: token
```

```
{
  "userId": "<adminUserId>"
}
```

Пример ответа

```
{
  "result": {
    "sessionId": "<sessionId>"
  },
  "success": true,
  "errorCode": 0
}
```

Полученный идентификатор сессии имеет срок действия, продлевающийся при обращениях к сервисам и истекающий в покое.

3.2. Добавление и изменение постоянного пользователя

3.2.1. Добавление пользователя (registerUser)

Следующий метод позволяет зарегистрировать в системе нового пользователя имеющего определенный ролевой контекст:

```
POST BaseUrl/tm-widgets/api/registerUser
accept: text/plain
Content-Type: application/json-patch+json
Authorization: <sessionId>
```

```
{
  "name": "string", //имя пользователя, произвольная строка, можно
использовать для идентификатора пользователя в системе-клиенте
  "roleContext": [{
    "VCS": {
      "id": "99229170299",
      "system": "SNILS",
      "fullName": "Крючкова Злата Степановна"
    }
  }, {
    "<Идентификатор ролевой схемы врача>": {
      "SNILS": "99229170299",
      "organization": "0b07be91-5f9f-4ca6-8ed5-16dd8cbac1f7"
    }
  }, {
    "<Идентификатор ролевой схемы врача>": {
      "SNILS": "99229170299",
      "organization": "083f7a14-729b-4671-97cb-09d08802bd8b"
    }
  }
]
}
```

В примере приведена регистрация пользователя врача-совместителя, работающего в двух организациях с использованием ролевого контекста для использования сервиса видеоконференцсвязи.

Пример ответа

```
{
  "success": true,
  "errorCode": 0,
  "message": "string",
  "stackTrace": "string",
  "result": {
    "userId": "<userId>",
    "name": "string",
    "roleContext": [ ... ]
  }
}
```

Полученное значение **userId** требуется сохранить в связи с добавляемым пользователем.

3.2.2. Чтение данных пользователя (GetUser)

Сервис ТМ позволяет получить данные по пользователям, ранее добавленным в сервис по идентификатору пользователя в сервисе (**userId**) по следующему методу:

```
GET BaseUrl/tm-widgets/api/user/<userId>
Authorization: <sessionId>
```

Пример ответа

```
{
  "result": {
    "id": "<userId>",

```

```

    "name": "string",
    "roleContexts": [
      {
        "id": "<rcId>",
        "userId": "<userId>",
        "roleContext":
          "[{\\"VCS\\":{\\"id\\":\\"99229170299\\",\\"system\\":\\"SNILS\\",\\"fullName\\":\\"Крючко
          ва Злата Степановна\\"}},{\\"<Идентификатор ролевой схемы
          врача>\\":{\\"SNILS\\":\\"99229170299\\",\\"organization\\":\\"0b07be91-5f9f-4ca6-
          8ed5-16dd8cba1f7\\"}},{\\"<Идентификатор ролевой схемы
          врача>\\":{\\"SNILS\\":\\"99229170299\\",\\"organization\\":\\"083f7a14-729b-4671-
          97cb-09d08802bd8b\\"}}]"
      }
    ]
  },
  "success": true,
  "errorCode": 0,
  "message": null,
  "stackTrace": null
}

```

roleContext сериализован в строку.

3.2.3. Изменение данных пользователя (updateUser)

Обновление данных пользователя сервиса ТМ возможно по идентификатору пользователя в сервисе (**userId**) следующим методом:

```

PUT BaseUrl/tm-widgets/api/updateUser/<userId>
accept: text/plain
Content-Type: application/json-patch+json
Authorization: <sessionId>

```

```

{
  "name": "string", //имя пользователя, произвольная строка, можно
  использовать для идентификатора пользователя в системе-клиенте
  "roleContext": [{
    "VCS": {
      "id": "99229170299",
      "system": "SNILS",
      "fullName": "Крючкова Злата Степановна"
    }
  }, {
    "<Идентификатор ролевой схемы врача>": {
      "SNILS": "99229170299",
      "organization": "09890547-e577-4733-9da5-e65cf0f16780"
    }
  }
]
}

```

Пример ответа

```

{
  "success": true,
  "errorCode": 0,
  "message": "string",
  "stackTrace": "string",
  "result": {
    "userId": "<userId>",
    "name": "string",
    "roleContext": [ ... ]
  }
}

```

3.3. Авторизация постоянных пользователей (sign-in)

Для совершения любых действий от имени авторизованного пользователя в сервисе расписаний в него следует передать идентификатор сессии пользователя, который должен быть предварительно получен с помощью следующего метода:

```
POST BaseUrl/tm-widgets/api/sign-in
accept: text/plain
Content-Type: application/json-patch+json
Authorization: token
```

```
{
  "userId": "<userId>"
}
```

Пример ответа

```
{
  "result": {
    "sessionId": "<userSessionId>"
  },
  "success": true,
  "errorCode": 0,
  "message": null,
  "stackTrace": null
}
```

Полученный идентификатор сессии имеет срок действия, продлевающийся при обращениях к сервисам и истекающий в покое.

3.4. Добавление пользователей-пациентов

3.4.1. Создание пациента (CreatePatient)

МИС МО предоставляет данные пациента посредством взаимодействия с сервисом платформы N3.Health «Индекс пациента».

Описание параметров запроса:

№	Название	Тип	Крат.	Описание
1.	misPatId	string	1..1	Идентификатор пациента в МИС <i>По справочнику: 1.2.643.5.1.13.2.7.100.7</i>
2.	moPatId	string	1..1	Идентификатор пациента в МО <i>По справочнику: 1.2.643.5.1.13.2.7.100.5</i>
3.	organization	string	0..1	Идентификатор МО в 1.2.643.2.69.1.1.1.64 Пример: <i>dfе3еес2-8а79-4921-9b58-0се03а5е6с10</i>
4.	firstName	string	1..1	Имя
5.	middleName	string	0..1	Отчество
6.	lastName	string	1..1	Фамилия

№	Название	Тип	Крат.	Описание
7.	gender	string	0..1	Пол 1 (либо М) – мужской пол 2 (либо Ж) – женский пол
8.	birthDate	string	1..1	Дата рождения Пример: 1985-11-20
9.	snils	string	0..1	СНИЛС
10.	inn	string	0..1	ИНН
11.	birthCertificate	string	0..1	Серия и номер свидетельства о рождении Пример: I-AK:123411
12.	dms	string	0..1	Полис ДМС Пример: 77-000543167
13.	dmsStart	string	0..1	Дата начала действия полиса ДМС Пример: 2024-11-20
14.	dmsEnd	string	0..1	Дата окончания действия полиса ДМС Пример: 2025-05-20
15.	passport	string	0..1	Серия и номер паспорта Пример: 6600:874511
16.	passportStart	string	0..1	Дата выдачи паспорта Пример: 2003-11-20
17.	polis	string	0..1	полис ОМС Пример: 7848710894041511
18.	email	string	0..1	Адрес электронной почты
19.	phone	string	0..1	Номер телефона
20.	address	string	0..1	Адрес

Пример запроса

POST **BaseUrl/tm-widgets/api/mpi/CreatePatient**

accept: text/plain

Content-Type: application/json-patch+json

Authorization: <userSessionId>

```
{
  "misPatId": "Test_20240927_1930",
  "moPatId": "Test_20240927_1930",
  "firstName": "Анна",
  "middleName": "Тестовна",
  "lastName": "Тестова",
  "gender": 2,
  "birthDate": "2002-04-10",
  "snils": "83121234301"
}
```

Редактирование данных по пациенту реализуется тем же методом с указанием параметров **misPatId** и **moPatId**, соответствующих запросу на создание пациента.

Описание параметров ответа:

№	Название	Тип	Крат.	Описание
1.	success	boolean	1..1	Успешность выполнения запроса
2.	errorCode	integer	1..1	Код ошибки <i>0 - без ошибок</i>
3.	message	string	0..1	Описание ошибки для пользователя <i>null - без ошибок</i>
4.	stackTrace	string	0..1	Техническое описание ошибки <i>null - без ошибок</i>
5.	result	string	0..1	Идентификатор пациента в сервисе (idMPI) «Индекс пациента» не указывается при неуспешном вызове

Пример ответа:

```
{
  "result": "147fb0f0-9802-419d-9a77-86a5aea355f8",
  "success": true,
  "errorCode": 0,
  "message": null,
  "stackTrace": null
}
```

Полученное значение в поле **result** далее будет обозначаться как idMPI

3.4.2. Получение данных о пациенте по идентификатору (SearchPatientById)

Сервис ТМ также позволяет получить данные о пациенте по его idMPI:

GET **BaseUrl/tm-widgets/api/mpi/SearchPatientById?patientId=<idMPI>**
Authorization: <userSessionId>

Описание параметров ответа:

№	Название	Тип	Крат.	Описание
1.	resourceType	string	1..1	Тип ресурса <i>Допустимые значения</i> Patient
2.	id	string	1..1	Идентификатор пациента в сервисе «Индекс пациента» <i>Пример</i> <i>8e4c76fa-aa65-45d5-9afc-79d01d1be623</i>
3.	meta	object	1..1	Метаданные объект-контейнер
3.1.	meta.extension	array	1..1	массив-контейнер
3.1.1.	(fhir-request-method) meta.extension[]	object	1..1	fhir-request-method объект-контейнер
3.1.1.1.	(fhir-request-method) meta.extension[].url	string	1..1	<i>Допустимые значения</i> fhir-request-method
3.1.1.2.	(fhir-request-method) meta.extension[].valueString	string	1..1	Метод запроса <i>Допустимые значения</i> POST

№	Название	Тип	Крат.	Описание
3.1.2.	(fhir-request-uri) meta. extension []	object	1..1	объект-контейнер
3.1.2.1.	(fhir-request-uri) meta.extension[]. url	string	1..1	Допустимые значения <i>fhir-request-uri</i>
3.1.2.2.	(fhir-request-uri) meta.extension[]. valueUri	string	1..1	Допустимые значения <i>Patient</i>
3.2.	meta. versionId	string	1..1	Идентификатор версии
3.3.	meta. lastUpdated	string	1..1	Время последнего обновления
4.	identifier	array	1..1	Идентификатор объекта массив-контейнер
4.1.	(в МО и МИС) identifier []	object	1..1	объект-контейнер
4.1.1.	(в МО и МИС) identifier[]. system	string	1..1	Идентификатор справочника Допустимые значения <i>urn:oid:1.2.643.5.1.13.2.7.100.5</i> – для МО <i>urn:oid:1.2.643.5.1.13.2.7.100.7</i> – для МИС
4.1.2.	(в МО и МИС) identifier[]. value	string	1..1	Идентификатор пациента
4.1.3.	(в МО и МИС) identifier[]. assigner	object	1..1	Участник информационного обмена - поставщик данных объект-контейнер
4.1.3.1.	(в МО и МИС) identifier[].assigner. display	string	1..1	Идентификатор поставщика данных по справочнику участников информационного обмена
4.2.	(документ пациента) identifier []	object	0..1	объект-контейнер
4.2.1.	(документ пациента) identifier[]. system	string	1..1	Идентификатор документа в справочнике видов документов Допустимые значения <i>urn:oid:1.2.643.2.69.1.1.1.6.14</i> – паспорт <i>urn:oid:1.2.643.2.69.1.1.1.6.223</i> – СНИЛС <i>urn:oid:1.2.643.2.69.1.1.1.6.228</i> – полис ОМС <i>urn:oid:1.2.643.2.69.1.1.1.6.240</i> – полис ДМС <i>urn:oid:1.2.643.2.69.1.1.1.6.249</i> – св-во о рождении <i>urn:oid:1.2.643.2.69.1.1.1.6.224</i> – ИНН
4.2.2.	(документ пациента) identifier[]. value	string	1..1	Номер документа <i>Пример паспорта</i> 6600:874511
4.2.3.	(документ пациента) identifier[]. assigner	object	1..1	Поставщик данных объект-контейнер
4.2.3.1.	(документ пациента) identifier[].assigner. display	string	1..1	Кем выдан
4.2.4.	(документ пациента) identifier[]. period	object	1..1	Период действия документа объект-контейнер
4.2.4.1.	(документ пациента) identifier[].period. start	string	1..1	Дата начала действия документа

№	Название	Тип	Крат.	Описание
4.2.4.2.	(документ пациента) identifier[.period.end	string	1..1	Дата окончания действия документа
5.	name	array	1..1	ФИО массив-контейнер
5.1.	name[]	object	1..1	объект-контейнер
5.1.1.	(Фамилия Отчество) name[.family	array	1..1	массив-контейнер
5.1.1.1.	(Фамилия) name[.family[0]	string	1..1	Фамилия
5.1.1.2.	(Отчество) name[.family[1]	string	0..1	Отчество
5.1.2.	name[.given	array	1..1	массив-контейнер
5.1.2.1.	name[.given[0]	string	1..1	Имя
6.	telecom	array	0..1	Контакты массив-контейнер
6.1.	telecom[]	object	0..*	объект-контейнер
6.1.1.	telecom[.system	string	1..1	Канал связи <i>Допустимые значения</i> phone fax email url other
6.1.2.	telecom[.value	string	1..1	Контакт
7.	gender	string	1..1	Пол <i>Допустимые значения</i> male female
8.	birthDate	string	1..1	Дата рождения <i>Пример</i> 1985-12-20
9.	address	array	0..1	Адрес массив-контейнер
9.1.	address[]	object	0..*	объект-контейнер
9.1.1.	address[.use	string	1..1	Тип адреса <i>Допустимые значения</i> home
9.1.2.	address[.text	string	1..1	Адрес
10.	managingOrganization	object	1..1	Медицинская организация создавшая запись о пациенте объект-контейнер
10.1.	managingOrganization .reference	string	1..1	Ссылка на МО Идентификатор МО в 1.2.643.2.69.1.1.1.64 <i>Пример</i> Organization/dfе3еес2-8а79-4921-9b58-0се03а5е6с10

Пример ответа:

```
{
  "resourceType": "Patient",
  "id": "3a316fa0-82e1-4c19-a9d6-035f98ef1f68",
  "meta": {
    "extension": [{
      "url": "fhir-request-method",
      "valueString": "POST"
    }, {
      "url": "fhir-request-uri",
      "valueUri": "Patient"
    }
  ],
  "versionId": "4a2b71f1-7fa7-4dc2-b126-3faf6ee573ca",
  "lastUpdated": "2023-11-22T18:13:14.5915285+03:00"
},
"identifier": [{
  "system": "urn:oid:1.2.643.5.1.13.2.7.100.5",
  "value": "c82c3ee1-8e5e-4d0a-8e40-c4721fd56720",
  "assigner": {
    "display": "1.2.643.2.69.1.2.116"
  }
}, {
  "system": "urn:oid:1.2.643.5.1.13.2.7.100.7",
  "value": "1db3c29e-2242-4720-a94e-b828af49e80b",
  "assigner": {
    "display": "1.2.643.2.69.1.2.116"
  }
}, {
  "system": "urn:oid:1.2.643.2.69.1.1.1.6.14",
  "value": "6600:874511",
  "assigner": {
    "display": "УФМС"
  }
}, {
  "system": "urn:oid:1.2.643.2.69.1.1.1.6.223",
  "value": "00094773141",
  "assigner": {
    "display": "ПФР"
  }
}, {
  "system": "urn:oid:1.2.643.2.69.1.1.1.6.228",
  "value": "7848710894041511"
}, {
  "system": "urn:oid:1.2.643.2.69.1.1.1.6.249",
  "value": "I-AK:123411"
}, {
  "system": "urn:oid:1.2.643.2.69.1.1.1.6.224",
  "value": "1234654311"
}, {
  "system": "urn:oid:1.2.643.2.69.1.1.1.6.240",
  "value": "77-0000012345",
  "period": {
    "start": "2024-11-26",
    "end": "2025-11-26"
  }
}
],
"name": [{
  "family": [
    "Решетников",
    "Андреевич"
  ]
}],
}
```

```
        "given": [
            "Виктор"
        ]
    },
],
"telecom": [{
    "system": "phone",
    "value": "79991112439"
}],
"gender": "male",
"birthDate": "1985-12-20",
"address": [{
    "use": "home",
    "text": "Какой-то текст тут"
}],
"managingOrganization": {
    "reference": "Organization/dfе3еес2-8а79-4921-9b58-0се03а5е6с10"
}
}
```

4. ОПИСАНИЕ МЕТОДОВ

Функциональность сервиса ТМ обеспечивается следующими методами:

Название	Описание
POST GetAvailableTransitions	Получение списка переходов доступных для создания заявки
GET GetSchema	Получение схемы данных, передаваемых при осуществлении перехода (опционально)
POST StartNewProcess	Создание новой заявки
POST GetTransitionAvailableProcesses	Получение списка всех заявок, доступных пользователю для осуществления перехода
GET ProcessContext	Получение контекста заявки
GET ProcessWithAvailableTransitions	Получение списка переходов доступных для заявки
POST MoveToStage	Редактирование/обновление заявки
POST GetReadAvailableProcesses	Получение списка доступных для просмотра заявок
GET Process	Получение заявки по идентификатору

В методах сервиса для передачи набора данных заявки используется параметр типа «Object» – processContext. Описание необходимого набора данных для каждого конкретного перехода по маршруту указывается в Спецификациях бизнес-процессов (маршрутов) обработки заявок.

4.1. Получение списка переходов доступных для создания заявки (GetAvailableTransitions)

Метод предоставляет перечень маршрутов и описания операций, в рамках которых возможно создавать заявки.

Описание параметров запроса:

№	Название	Тип	Крат.	Описание
1.	workflowFilter	object	1..1	объект-контейнер
2.	skip	integer	0..1	Количество пропущенных элементов в выборке
3.	take	integer	0..1	Количество элементов в списке

Пример запроса

```
POST BaseUrl/tm-widgets/api/TmCoreInteraction/GetAvailableTransitions  
accept: text/plain  
Content-Type: application/json-patch+json  
Authorization: <userSessionId>
```

```
{  
  "WorkflowFilter": {
```

```

    "Id": "f24be524-fb0a-42c3-a108-bcee78081792",
    "Name": "Консультация пациенту",
    "Metadata": {}
  },
  "Skip": 0,
  "Take": 10
}

```

`workflowFilter` – объект для фильтрации по маршруту. Все поля являются опциональными (не обязательными), например:

- `Id` – идентификатор маршрута;
- `Name` – название маршрута (точное совпадение);
- `Metadata` - фильтр по объекту метаданных маршрута.

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

Описание параметров ответа:

№	Название	Тип	Крат.	Описание
1.	success	boolean	1..1	Успешность обработки запроса
2.	errorCodes	integer	1..1	Код ошибки
3.	message	string	1..1	Сообщение об ошибке
4.	stackTrace	string	1..1	Техническая информация об ошибке
5.	result	object	1..1	объект-контейнер (не указывается при неуспешном вызове)
5.1.	result.result	array	1..1	массив-контейнер
5.1.1.	result.result[]	object	0..*	Результат по маршруту объект-контейнер
5.1.1.1.	result.result[].workflowId	string	1..1	Идентификатор маршрута
5.1.1.2.	result.result[].transitions	array	1..1	Переходы массив-контейнер
5.1.1.2.1.	result.result[].transitions[]	object	0..*	объект-контейнер
5.1.1.2.1.1.	result.result[].transitions[].id	string	1..1	Идентификатор перехода
5.1.1.2.1.2.	result.result[].transitions[].name	string	1..1	Название перехода
5.1.1.2.1.3.	result.result[].transitions[].fromStageId	string	0..1	Идентификатор состояния, с которого производится переход
5.1.1.2.1.4.	result.result[].transitions[].toStageId	string	1..1	Идентификатор состояния, на который производится переход
5.1.1.2.1.5.	result.result[].transitions[].schemaId	string	1..1	Идентификатор схемы передаваемых данных
5.1.1.2.1.6.	result.result[].transitions[].validatorIds	array	1..1	Проверки массив-контейнер
5.1.1.2.1.6.1.	result.result[].transitions[].validatorIds[]	string	0..*	Идентификатор проверки

№	Название	Тип	Крат.	Описание
5.1.1.2.1.7.	result.result[].transitions[].callbackIds	array	1..1	Внешние вызовы массив-контейнер
5.1.1.2.1.7.1.	result.result[].transitions[].callbackIds[]	string	0..*	Идентификатор вызова
5.1.1.2.1.8.	result.result[].transitions[].roleSchemaIds	array	1..1	Ролевые схемы массив-контейнер
5.1.1.2.1.8.1.	result.result[].transitions[].roleSchemaIds[]	string	1..*	Идентификатор схемы
5.1.1.3.	result.result[].workflowIsConstruction	boolean	1..1	Статус маршрута
5.1.1.4.	result.result[].workflowIsDisabled	boolean	1..1	Метка удаления маршрута
5.1.1.5.	result.result[].workflowName	string	1..1	Название маршрута
5.1.1.6.	result.result[].workflowMetadata	object	1..1	метаданные маршрута объект-контейнер
5.2.	result.total	integer	1..1	количество элементов в выдаче

Пример ответа

```
{
  "result": {
    "result": [
      {
        "workflowIsConstruction": false,
        "workflowIsDisabled": false,
        "workflowName": "Консультация пациенту",
        "workflowMetadata": {},
        "workflowId": "f24be524-fb0a-42c3-a108-bcee78081792",
        "transitions": [
          {
            "id": "0581865f-67db-4aef-9358-6227fd7e878b",
            "name": "Создать заявку по согласованному времени",
            "fromStageId": null,
            "toStageId": "b4065ae9-274f-4cf5-9915-cedeeaaf7559",
            "schemaId": "bc55ae38-1f92-49ca-9281-7e6dca493a4f",
            "validatorIds": [
              "4c727ebe-7822-45e5-b242-e1bf7d74d746"
            ],
            "callbackIds": [],
            "roleSchemaIds": [
              "5ff16ba7-9edd-41a4-bd06-2ee33cfd4597"
            ],
            "isDeleted": false,
            "message": null
          }
        ]
      }
    ],
    "total": 1
  },
  "success": true,
  "errorCode": 0,
  "message": null,
  "stackTrace": null
}
```

4.2. Получение схемы данных, передаваемых при осуществлении перехода (GetSchema)

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

данных, необходимых на каждом переходе содержатся в спецификации каждого конкретного маршрута.

Пример запроса

```
GET BaseUrl/tm-widgets/api/TmCoreInteraction/GetSchema?schemaId=<Ид. схемы данных>
Authorization: <userSessionId>
```

Описание параметров ответа:

№	Название	Тип	Крат.	Описание
1.	success	boolean	1..1	Успешность обработки запроса
2.	errorCodes	integer	1..1	Код ошибки
3.	message	string	1..1	Сообщение об ошибке
4.	stackTrace	string	1..1	Техническая информация об ошибке
5.	result	object	1..1	объект-контейнер
5.1.	result.id	string	1..1	Идентификатор схемы
5.2.	result.name	string	1..1	Название схемы
5.3.	result.type	integer	1..1	Тип схемы <i>Допустимые значения: 0 1 2 3 4</i>
5.4.	result.areaId	string	1..1	Идентификатор предметной области
5.5.	result.areaName	string	1..1	Название предметной области
5.6.	result.schema	object	1..1	JSON-схема объект-контейнер
5.7.	result.isDisabled	boolean	1..1	Признак неактивности

Пример ответа

```
{
  "result": {
    "id": "b7d7cc41-44ab-432f-a831-12fe247174db",
    "name": "Заключение",
    "type": 3,
    "areaId": "50752145-1d1b-42cc-b052-2892a84485ed",
    "areaName": "Телемедицина",
    "schema": {
      "type": "object",
      "$schema": "http://json-schema.org/draft-04/schema#",
      "required": [
        "diagnosticReport"
      ],
      "properties": {
        "appointment": {
          "type": "object",
          "title": "Время (фактическое, желаемое) обработки заявки",
          "required": [
            "end",
            "start"
          ],
          "properties": {
            "start": {
              "type": "string",
              "title": "Время начала"
            }
          }
        }
      }
    }
  }
}
```

```

    }
    "end": {
      "type": "string",
      "title": "Время завершения"
    },
    ...
  },
  "isDisabled": false
},
"success": true,
"errorCode": 0,
"message": null,
"stackTrace": null
}

```

4.3. Создание новой заявки (StartNewProcess)

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

Описание параметров запроса:

№	Название	Тип	Крат.	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 соответствующий json-схеме для указанного перехода. Схема может быть получена с помощью последовательного применения двух запросов: GET /GetTransition/<transitionId> \$.result.schemaId из ответа использовать в следующем запросе GET /GetSchema/<schemaId> Также схема указана в Спецификациях бизнес-процессов (маршрутов) обработки заявок

Пример запроса

```

POST BaseUrl/tm-widgets/api/TmCoreInteraction/StartNewProcess
Authorization : sessionId
Content-Type: application/json-patch+json
Authorization: <userSessionId>

```

```

{
  "workflowId": "<Идентификатор маршрута>",
  "name": "<Человекочитаемое имя заявки>",
  "initialTransitionId": "<Идентификатор стартового перехода>",
  "processContext": {
    <Контекст перехода в соответствии со схемой schemaId>
  }
}

```

Пример передачи тестовой заявки:

```

POST BaseUrl/tm-widgets/api/TmCoreInteraction/StartNewProcess
Authorization : sessionId
Content-Type: application/json-patch+json
Authorization: <userSessionId>

```

```

{

```



```

"initialTransitionId": "0581865f-67db-4aef-9358-6227fd7e878b",
"name": "Новая заявка 06.02.2025 12:58:14",
"workflowId": "f24be524-fb0a-42c3-a108-bcee78081792",
"processContext": {
  "serviceRequest": {
    "performerOrganization": "11ea52be-1543-4977-8f9c-038521fd3d9e",
    "healthCareService": "f60e4b0f-f8c2-49e8-bb1f-7db15146b49b",
    "category": "176",
    "complaints": "Жалобы пациента",
    "requesterOrganization": "11ea52be-1543-4977-8f9c-038521fd3d9e"
  },
  "diagnosticReport": {
    "idCaseAidType": "A07.16.006"
  },
  "locationType": "2",
  "performerPractitioner": {
    "identityDocument": {
      "system": "223",
      "code": "87857464658"
    },
    "fullName": "Мещерская Ольга Викторовна",
    "position": "195",
    "specialty": "1",
    "department": "f2ea529f-b5f1-4c97-a1f4-339a36c243b6"
  },
  "medDocks": [],
  "appointment": {
    "start": "2025-02-07T13:00:00+03:00",
    "end": "2025-02-07T14:00:00+03:00"
  },
  "performerCondition": {
    "codeМКВ": "O30.0"
  },
  "requesterPractitioner": {
    "identityDocument": {
      "system": "223",
      "code": "87857464658"
    },
    "fullName": "Мещерская Ольга Викторовна",
    "position": "195",
    "specialty": "1",
    "department": "f2ea529f-b5f1-4c97-a1f4-339a36c243b6"
  },
  "patient": {
    "idMPI": "4e42bb72-12ab-4e08-91de-91f4e922faa6",
    "fullName": "Тестовая Анна Тестовна",
    "gender": "1",
    "birthDate": "1984-10-27",
    "identityDocument": {
      "code": "03356688270",
      "system": "urn:oid:1.2.643.2.69.1.1.1.6.223"
    }
  }
}
}
}

```

При формировании структуры объекта processContext следует учитывать, что объект для конкретного маршрута будет отличаться от приведенного выше примера. Уточнение требуемой структуры и обязательность полей на каждом отдельно взятом маршруте приведено в спецификации бизнес-процессов (маршрутов) обработки заявок.

В случае работы с заявками по открытому расписанию (на маршрутах врач-пациент, врач-врач), сведения об услугах, исполнителях и слотах требуется получать из сервиса Расписаний и записи. Инструкция по интеграции с сервисом размещена по адресу <https://api.n3health.ru/> в разделе «Сервис Записи».

Описание параметров ответа:

№	Название	Тип	Крат.	Описание
1.	success	boolean	1..1	Успешность обработки запроса
2.	errorCodes	integer	1..1	Код ошибки
3.	message	string	1..1	Сообщение об ошибке
4.	stackTrace	string	1..1	Техническая информация об ошибке
5.	workflowId	string	1..1	Идентификатор маршрута (не указывается при неуспешном вызове)
6.	processId	string	1..1	Идентификатор заявки (не указывается при неуспешном вызове)
7.	stageId	string	1..1	Идентификатор текущего статуса заявки (не указывается при неуспешном вызове)
8.	currentTransition	string	1..1	Текущий переход (не указывается при неуспешном вызове)
9.	humanFriendlyId	string	1..1	Человекочитаемый номер заявки (не указывается при неуспешном вызове)
10.	validationResults	array	1..1	Ошибки валидации (null при успешном вызове) массив-контейнер
10.1.	validationResults[]	undefined	0..*	Идентификаторы ошибок

Пример ответа

```
{
  "workflowId": "f24be524-fb0a-42c3-a108-bcee78081792",
  "processId": "9da43288-1169-4c13-88dc-323826b3167f",
  "stageId": "b4065ae9-274f-4cf5-9915-ceedeaaaf7559",
  "currentTransition": null,
  "humanFriendlyId": "TMC02257475328",
  "validationResults": null,
  "success": true,
  "errorCode": 0,
  "message": null,
  "stackTrace": null
}
```

4.4. Получение списка всех заявок, доступных пользователю для осуществления перехода (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:

№	Название	Тип	Крат.	Описание
1.	workflowFilter	object	0..1	

№	Название	Тип	Крат.	Описание
1.1.	workflowFilter.id	string	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 описаны в спецификации бизнес-процессов (маршрутов) обработки заявок. <ul style="list-style-type: none"> Name – название маршрута (строка) Metadata – фильтр по объекту метаданных маршрута
1.2.	workflowFilter.name	string	0...1	
1.3.	workflowFilter.area	string	0...1	
1.4.	workflowFilter.description	string	0...1	
1.5.	workflowFilter.metadata	object	0...1	
2.	processFilter	object	0...1	Параметры фильтра по схеме описания метаданных заявки. ProcessFilter – фильтр заявок непосредственно по атрибутам и метаданным самой заявки. Атрибуты заявки присваиваются ей при создании и обновлении. Метаданные заявки генерируются при создании и каждом обновлении по соответствующей схеме (см. метод получения схем данных).
2.1.	processFilter.name	string	0...1	
2.2.	processFilter.humanFriendlyId	string	0...1	
2.3.	processFilter.workflow	string	0...1	
2.4.	processFilter.metadata	object	0...1	
2.5.	processFilter.created	string	0...1	
2.6.	processFilter.updated	string	0...1	
3.	stageFilter	array	0...1	Фильтр по идентификаторам текущих статусов заявок
3.1.	stageFilter[]	string	0..*	Идентификаторы текущих заявок
4.	skip	integer	0...1	Количество пропущенных элементов в выборке
5.	take	integer	0...1	Количество элементов в списке
6.	orderingField	string	0...1	Поле сортировки
7.	descendingOrder	boolean	0...1	Направление сортировки
8.	businessStatusCodes	array	0...1	Массив статусов заявок
8.1.	businessStatusCodes[]	string	0..*	Идентификатор статусов заявок
9.	selectedRole	string	0..1	Идентификатор пользовательской роли

Пример запроса

POST **BaseUrl/tm-widgets/api/TmCoreInteraction/GetTransitionAvailableProcesses**
 accept: text/plain
 Content-Type: application/json-patch+json
 Authorization: <userSessionId>

```
{
  "selectedRole": "5ff16ba7-9edd-41a4-bd06-2ee33cfd4597",
  "workflowFilter": {},
  "processFilter": {
    "name": "Человекочитаемое имя заявки",
    "created": [
      "0001-01-01T02:30:17+02:30",
      "2025-02-06T16:57:44+03:00"
    ],
    "updated": [
      "0001-01-01T02:30:17+02:30",
      "2025-02-06T16:57:44+03:00"
    ],
    "workflow": "f24be524-fb0a-42c3-a108-bcee78081792"
  },
}
```

```

"stageFilter": [
  "d4e24211-d8f2-4cda-831c-519c80f0343e"
],
"descendingOrder": "true",
"orderingField": "updated",
"skip": 0,
"take": 20
}

```

Описание параметров ответа:

№	Название	Тип	Крат.	Описание
1.	success	boolean	1..1	Успешность обработки запроса
2.	errorCodes	integer	1..1	Код ошибки
3.	message	string	1..1	Сообщение об ошибке
4.	stackTrace	string	1..1	Техническая информация об ошибке
5.	result	object	1..1	объект-контейнер (не указывается при неуспешном вызове)
5.1.	result.result	array	1..1	массив-контейнер
5.1.1.	result.result[]	object	1..1	объект-контейнер
5.1.1.1.	result.result[].processId	string	1..1	Системный идентификатор заявки
5.1.1.2.	result.result[].processHumanFriendlyId	string	1..1	Человекочитаемый идентификатор заявки
5.1.1.3.	result.result[].metadata	object	1..1	Метаданные маршрута
5.1.1.4.	result.result[].scopedMetadata	object	1..1	Расшифрованные метаданные. В состав включаются все поля метаданных у которых в json-схеме указан атрибут "displayInUI": true
5.1.1.5.	result.result[].currentStageId	string	1..1	Идентификатор текущего статуса заявки
5.1.1.6.	result.result[].currentStage	string	1..1	Текущий статус заявки
5.1.1.7.	result.result[].workflowId	string	1..1	Идентификатор маршрута
5.1.1.8.	result.result[].workflowName	string	1..1	Название маршрута
5.1.1.9.	result.result[].processName	string	1..1	Название заявки
5.1.1.10.	result.result[].created	string	1..1	Отметка времени создания заявки
5.1.1.11.	result.result[].updated	string	1..1	Отметка времени последнего изменения
5.1.1.12.	result.result[].businessStatus	object	1..1	Бизнес статус заявки объект-контейнер
5.1.1.12.1.	result.result[].businessStatus.system	string	1..1	OID справочника состояний <i>urn:oid:1.2.643.2.69.1.1.148.2</i>
5.1.1.12.2.	result.result[].businessStatus.code	string	1..1	Код состояния <i>Пример: 17</i>
5.1.1.13.	result.result[].transitions	array	1..1	Переходы массив-контейнер
5.1.1.13.1.	result.result[].transitions[]	object	0..*	объект-контейнер
5.1.1.13.1.1.	result.result[].transitions[].id	string	1..1	Идентификатор перехода

№	Название	Тип	Крат.	Описание
5.1.1.13.1.2.	result.result[].transitions[].name	string	1..1	Наименование перехода
5.1.1.13.1.3.	result.result[].transitions[].fromStageId	string	0..1	Идентификатор состояния заявки, из которого можно совершить переход. При отсутствии, переход используется для создания заявки
5.1.1.13.1.4.	result.result[].transitions[].toStageId	string	1..1	Идентификатор состояния заявки, на который будет осуществлен переход
5.1.1.13.1.5.	result.result[].transitions[].schemaId	string	0..1	Идентификатор схемы данных необходимых для выполнения операции
5.1.1.13.1.6.	result.result[].transitions[].validatorIds	array	1..1	Проверки массив-контейнер
5.1.1.13.1.6.1.	result.result[].transitions[].validatorIds[]	string	0..*	Идентификатор проверки
5.1.1.13.1.7.	result.result[].transitions[].callbackIds	array	1..1	Внешние вызовы массив-контейнер
5.1.1.13.1.7.1.	result.result[].transitions[].callbackIds[]	string	0..*	Идентификатор внешнего вызова
5.1.1.13.1.8.	result.result[].transitions[].roleSchemaIds	array	1..1	массив-контейнер
5.1.1.13.1.8.1.	result.result[].transitions[].roleSchemaIds[]	string	1..*	Ролевые схемы
5.2.	result.total	integer	1..1	количество элементов в выдаче

4.5. Получение контекста заявки (ProcessContext)

Метод предназначен для получения данных (объекта контекста) заявки.

Пример запроса

```
GET BaseUrl/tm-widgets/api/TmCoreInteraction/ProcessContext?processId=<Ид. процесса>
Authorization: <userSessionId>
```

Описание параметров ответа:

№	Название	Тип	Крат.	Описание
1.	success	boolean	1..1	Успешность обработки запроса
2.	errorCodes	integer	1..1	Код ошибки
3.	message	string	1..1	Сообщение об ошибке
4.	stackTrace	string	1..1	Техническая информация об ошибке
5.	result	object	1..1	Контекст заявки объект-контейнер

Пример ответа

```

{
  "result": {
    "patient": {
      "idMPI": "4e42bb72-12ab-4e08-91de-91f4e922faa6",
      "gender": "1",
      "fullName": "Тестовая Анна Тестовна",
      "birthDate": "1984-10-27",
      "identityDocument": {
        "code": "03356688270",
        "system": "urn:oid:1.2.643.2.69.1.1.1.6.223"
      }
    },
    "messages": {
      "paymentRequired": "Оплата не требуется",
      "reasonForNoConclusion": "1"
    },
    "appointment": {
      "end": "2025-02-07T14:00:00+03:00",
      "start": "2025-02-07T13:00:00+03:00"
    },
    "locationType": "2",
    "serviceRequest": {
      "complaints": "Жалобы пациента",
      "healthCareService": "f60e4b0f-f8c2-49e8-bb1f-7db15146b49b",
      "performerOrganization": "11ea52be-1543-4977-8f9c-038521fd3d9e",
      "requesterOrganization": "11ea52be-1543-4977-8f9c-038521fd3d9e"
    },
    "performerCondition": {
      "codeМКВ": "O30.0"
    },
    ...
  },
  "success": true,
  "errorCode": 0,
  "message": null,
  "stackTrace": null
}

```

4.6. Получение списка переходов доступных для заявки (ProcessWithAvailableTransitions)

Метод предоставляет описание заявки со списком переходов, доступных пользователю в данной заявке, учитывая текущий статус заявки.

Пример запроса

```

GET BaseUrl/tm-widgets/api/TmCoreInteraction/ProcessWithAvailableTransitions?
processId=<Идентификатор процесса>
Authorization: <userSessionId>

```

Описание параметров ответа:

№	Название	Тип	Крат.	Описание
1.	success	boolean	1..1	Успешность обработки запроса
2.	errorCodes	integer	1..1	Код ошибки
3.	message	string	1..1	Сообщение об ошибке
4.	stackTrace	string	1..1	Техническая информация об ошибке
5.	result	object	1..1	объект-контейнер (не указывается при неуспешном вызове)

№	Название	Тип	Крат.	Описание
5.1.	result. id	string	1..1	Идентификатор заявки
5.2.	result. name	string	1..1	Наименование заявки
5.3.	result. workflowId	string	1..1	Идентификатор маршрута
5.4.	result. metadataId	string	1..1	Идентификатор метаданных
5.5.	result. currentStageId	string	1..1	Идентификатор текущего статуса заявки
5.6.	result. currentStageName	string	1..1	Наименование текущего статуса заявки
5.7.	result. created	string	1..1	Отметка времени создания заявки
5.8.	result. updated	string	1..1	Отметка времени последнего изменения заявки
5.9.	result. businessStatus	object	1..1	Бизнес статус заявки объект-контейнер
5.9.1.	result.businessStatus. system	string	1..1	OID справочника состояний <i>Пример</i> <i>urn:oid:1.2.643.2.69.1.1.1.148.2</i>
5.9.2.	result.businessStatus. code	string	1..1	Код состояния <i>Пример: 17</i>
5.10.	result. humanFriendlyId	string	1..1	Человекочитаемый идентификатор заявки
5.11.	result. availableTransitions	array	1..1	массив-контейнер
5.11.1.	result. availableTransitions[]	object	0..1	Переходы объект-контейнер
5.11.1.1.	result.availableTransitions[]. id	string	1..1	Идентификатор перехода
5.11.1.2.	result.availableTransitions[]. name	string	1..1	Наименование перехода
5.11.1.3.	result.availableTransitions[]. fromStageId	string	1..1	Идентификатор состояния, с которого производится переход
5.11.1.4.	result.availableTransitions[]. toStageId	string	1..1	Идентификатор состояния, на который производится переход
5.11.1.5.	result.availableTransitions[]. schemaId	string	1..1	Идентификатор схемы передаваемых данных
5.11.1.6.	result.availableTransitions[]. validatorIds	array	1..1	Проверки массив-контейнер
5.11.1.6.1.	result.availableTransitions[]. validatorIds[]	string	0..*	Идентификатор проверки
5.11.1.7.	result.availableTransitions[]. callbackIds	array	1..1	Внешние вызовы массив-контейнер
5.11.1.7.1.	result.availableTransitions[]. callbackIds[]	string	0..*	Идентификатор внешнего вызова
5.11.1.8.	result.availableTransitions[]. roleSchemaIds	array	1..1	массив-контейнер
5.11.1.8.1.	result.availableTransitions[]. roleSchemaIds[]	string	1..*	Рольевые схемы
5.12.	result. metaData	object	1..1	Метаданные заявки.

№	Название	Тип	Крат.	Описание
5.13.	result.scopedMetadata	object	1..1	Расшифрованные метаданные объект-контейнер
5.14.	result.statusName	string	1..1	Название текущего статуса заявки
5.15.	result.primarySchemaId	string	1..1	Идентификатор схемы данных предметной области

Пример ответа

```
{
  "result": {
    "id": "9da43288-1169-4c13-88dc-323826b3167f",
    "name": "Новая заявка 06.02.2025 12:58:14",
    "workflowId": "f24be524-fb0a-42c3-a108-bcee78081792",
    "metadataId": "543c93d4-cdf1-45d2-b71c-b9283b09417a",
    "currentStageId": "9ac1c071-f4db-465f-913f-d65f8f6a6cd8",
    "currentStageName": "Подготовить заключение",
    "created": "2025-02-06T13:38:22.207803+03:00",
    "updated": "2025-02-06T15:33:14.83948+03:00",
    "businessStatus": null,
    "humanFriendlyId": "TMC02257475328",
    "availableTransitions": [
      {
        "id": "52843faf-0c04-4c9f-8c31-4d6d8600d93c",
        "name": "Завершить консультацию и подготовить заключение",
        "fromStageId": "9ac1c071-f4db-465f-913f-d65f8f6a6cd8",
        "toStageId": "35d447df-3f9f-49bd-8b01-0b5e70de3f52",
        "schemaId": "b7d7cc41-44ab-432f-a831-12fe247174db",
        "validatorIds": [
          "4dae198c-11f7-463c-bcdd-d26da2ccb9e5"
        ],
        "callbackIds": [],
        "roleSchemaIds": [
          "5ff16ba7-9edd-41a4-bd06-2ee33cfd4597"
        ],
        "isDeleted": false,
        "message": null
      },
      ...
    ],
    "metaData": { ... },
    "scopedMetadata": { ... },
    "statusName": "Подготовить заключение",
    "primarySchemaId": "815f2f6f-0e44-4975-9a47-7b75d4a90ae1"
  },
  "success": true,
  "errorCode": 0,
  "message": null,
  "stackTrace": null
}
```

4.7. Редактирование/обновление заявки (MoveToStage)

Для редактирования или обновления данных заявки в сервисе используется метод POST MoveToStage. В базовом назначении метод позволяет осуществить запрос перехода заявки в новое состояние согласно указанному правилу перехода.

Помимо редактирования заявки, запрос метода используется для обновления состояния заявки в любых других операциях доступных по заявке.

Описание параметров запроса:

№	Название	Тип	Крат.	Описание
1.	processId	string	1..1	Идентификатор заявки
2.	transitionId	string	1..1	Идентификатор перехода
3.	processContext	undefined	1..1	Набор данных соответствующий json-схеме для указанного перехода. Схема может быть получена с помощью последовательного применения двух запросов: GET /GetTransition/<transitionId> \$.result.schemaId из ответа использовать в следующем запросе GET /GetSchema/<schemaId> поле тела ответа \$.result.schema содержит искомую JSON-схему Также схема указана в Спецификациях бизнес-процессов (маршрутов) обработки заявок.

Пример запроса

```
POST BaseUrl/tm-widgets/api/TmCoreInteraction/MoveToStage
accept: text/plain
Content-Type: application/json-patch+json
Authorization: <userSessionId>
```

```
{
  "TransitionId": "9d7631e4-1f3b-4f75-a437-b8894ee9938a",
  "ProcessId": "9da43288-1169-4c13-88dc-323826b3167f",
  "ProcessContext": {
    "messages": {
      "reasonForNoConclusion": "Заклучение не требуется пациенту"
    }
  }
}
```

Описание параметров ответа:

№	Название	Тип	Крат.	Описание
1.	success	boolean	1..1	Успешность обработки запроса
2.	errorCodes	integer	1..1	Код ошибки
3.	message	string	1..1	Сообщение об ошибке
4.	stackTrace	string	1..1	Техническая информация об ошибке
5.	workflowId	string	1..1	Идентификатор маршрута (не указывается при неуспешном вызове)
6.	processId	string	1..1	Идентификатор заявки (не указывается при неуспешном вызове)
7.	stageId	string	1..1	Идентификатор текущего статуса заявки (не указывается при неуспешном вызове)
8.	currentTransition	string	1..1	Текущий переход (не указывается при неуспешном вызове)
9.	humanFriendlyId	string	1..1	Человекочитаемый номер заявки (не указывается при неуспешном вызове)

№	Название	Тип	Крат.	Описание
10.	validationResults	array	1..1	Ошибки валидации (null при успешном вызове) массив-контейнер
10.1.	validationResults[]	undefined	0..*	Идентификаторы ошибок

Пример ответа

```
{
  "workflowId": "f24be524-fb0a-42c3-a108-bcee78081792",
  "processId": "9da43288-1169-4c13-88dc-323826b3167f",
  "stageId": "b4065ae9-274f-4cf5-9915-ceedeaaaf7559",
  "currentTransition": null,
  "humanFriendlyId": "TMC02257475328",
  "validationResults": null,
  "success": true,
  "errorCode": 0,
  "message": null,
  "stackTrace": null
}
```

4.8. Получение списка доступных для просмотра заявок (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:

№	Название	Тип	Крат.	Описание
1.	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 описаны в спецификации бизнес-процессов (маршрутов) обработки заявок. • Name – название маршрута (строка) • Metadata – фильтр по объекту метаданных маршрута
1.1.	workflowFilter.id	string	0..1	
1.2.	workflowFilter.name	string	0..1	
1.3.	workflowFilter.area	string	0..1	
1.4.	workflowFilter.description	string	0..1	
1.5.	workflowFilter.metadata	object	0..1	
2.	processFilter	object	0..1	Параметры фильтра по схеме описания метаданных заявки. ProcessFilter – фильтр заявок непосредственно по атрибутам и метаданным самой заявки. Атрибуты заявки присваиваются ей при создании и обновлении. Метаданные заявки генерируются при создании и каждом обновлении по соответствующей схеме (см. метод получения схем данных).
2.1.	processFilter.name	string	0..1	
2.2.	processFilter.humanFriendlyId	string	0..1	
2.3.	processFilter.workflow	string	0..1	
2.4.	processFilter.metadata	object	0..1	
2.5.	processFilter.created	string	0..1	
2.6.	processFilter.updated	string	0..1	

№	Название	Тип	Крат.	Описание
3.	stageFilter	array	0...1	Фильтр по идентификаторам текущих статусов заявок
3.1.	stageFilter[]	string	0..*	Идентификаторы текущих заявок
4.	skip	integer	0...1	Количество пропущенных элементов в выборке
5.	take	integer	0...1	Количество элементов в списке
6.	orderingField	string	0...1	Поле сортировки
7.	descendingOrder	boolean	0...1	Направление сортировки
8.	businessStatusCodes	array	0...1	Массив статусов заявок
8.1.	businessStatusCodes[]	string	0..*	Идентификатор статусов заявок
9.	selectedRole	string	0..1	Идентификатор пользовательской роли

Пример запроса

POST **BaseUrl/tm-widgets/api/TmCoreInteraction/GetReadAvailableProcesses**

accept: text/plain

Content-Type: application/json-patch+json

Authorization: <userSessionId>

```
{
  "sessionId": "9b73099e-2b92-4bc5-96f0-389a8cef9b68"
  "selectedRole": "5ff16ba7-9edd-41a4-bd06-2ee33cfd4597",
  "workflowFilter": {},
  "processFilter": {
    "name": "Человекочитаемое имя заявки",
    "created": [
      "0001-01-01T02:30:17+02:30",
      "2025-02-06T17:05:03+03:00"
    ],
    "updated": [
      "0001-01-01T02:30:17+02:30",
      "2025-02-06T17:05:03+03:00"
    ],
    "workflow": "f24be524-fb0a-42c3-a108-bcee78081792"
  },
  "stageFilter": [
    "d4e24211-d8f2-4cda-831c-519c80f0343e"
  ],
  "descendingOrder": "true",
  "orderingField": "updated",
  "skip": 0,
  "take": 20
}
```

Описание параметров ответа:

№	Название	Тип	Крат.	Описание
1.	success	boolean	1..1	Успешность обработки запроса
2.	errorCodes	integer	1..1	Код ошибки
3.	message	string	1..1	Сообщение об ошибке
4.	stackTrace	string	1..1	Техническая информация об ошибке
5.	result	object	1..1	объект-контейнер (не указывается в неуспешном вызове)
5.1.	result.result	array	1..1	массив-контейнер

№	Название	Тип	Крат.	Описание
5.1.1.	result.result[]	object	1..1	объект-контейнер
5.1.1.1.	result.result[].processId	string	1..1	Системный идентификатор заявки
5.1.1.2.	result.result[].processHumanFriendlyId	string	1..1	Человекочитаемый идентификатор заявки
5.1.1.3.	result.result[].metadata	object	1..1	Метаданные маршрута
5.1.1.4.	result.result[].scopedMetadata	object	1..1	Расшифрованные метаданные в состав расшифрованных метаданных включаются все поля метаданных у которых в json-схеме указан атрибут "displayInUI": true
5.1.1.5.	result.result[].currentStageId	string	1..1	Идентификатор текущего статуса заявки
5.1.1.6.	result.result[].currentStage	string	1..1	Текущий статус заявки
5.1.1.7.	result.result[].workflowId	string	1..1	Идентификатор маршрута
5.1.1.8.	result.result[].workflowName	string	1..1	Название маршрута
5.1.1.9.	result.result[].processName	string	1..1	Название заявки
5.1.1.10.	result.result[].created	string	1..1	Отметка времени создания заявки
5.1.1.11.	result.result[].updated	string	1..1	Отметка времени последнего изменения заявки
5.1.1.12.	result.result[].businessStatus	object	1..1	Бизнес статус заявки объект-контейнер
5.1.1.12.1.	result.result[].businessStatus.system	string	1..1	OID справочника состояний <i>urn:oid:1.2.643.2.69.1.1.1.148.2</i>
5.1.1.12.2.	result.result[].businessStatus.code	string	1..1	Код состояния <i>Пример: 17</i>
5.2.	result.total	integer	1..1	количество элементов в выдаче

4.9. Получение заявки по идентификатору (Process)

Метод предоставляет сведения о заявке на ее текущем статусе.

Пример запроса

```
GET BaseUrl/tm-widgets/api/TmCoreInteraction/Process?processId=<Ид. процесса>
Authorization: <userSessionId>
```

Описание параметров ответа:

№	Название	Тип	Крат.	Описание
1.	success	boolean	1..1	Успешность обработки запроса
2.	errorCodes	integer	1..1	Код ошибки
3.	message	string	1..1	Сообщение об ошибке
4.	stackTrace	string	1..1	Техническая информация об ошибке

№	Название	Тип	Крат.	Описание
5.	result	object	1..1	объект-контейнер (не указывается при неуспешном вызове)
5.1.	result.id	string	1..1	массив-контейнер
5.2.	result.name	string	1..1	объект-контейнер
5.3.	result.workflowId	string	1..1	Идентификатор маршрута
5.4.	result.metadataId	string	1..1	Идентификатор метаданных
5.5.	result.currentStageId	string	1..1	Идентификатор текущего статуса заявки
5.6.	result.currentStageName	string	1..1	Наименование текущего статуса заявки
5.7.	result.created	string	1..1	Отметка времени создания заявки
5.8.	result.updated	string	1..1	Отметка времени последнего изменения заявки
5.9.	result.businessStatus	object	1..1	Бизнес статус заявки объект-контейнер
5.9.1.	result.businessStatus.system	string	1..1	OID справочника состояний <i>Пример: urn:oid:1.2.643.2.69.1.1.1.148.2</i>
5.9.2.	result.businessStatus.code	string	1..1	Код состояния <i>Пример: 17</i>
5.10.	result.humanFriendlyId	string	1..1	Человекочитаемый номер заявки (не указывается при неуспешном вызове)
5.11.	result.metadata	object	1..1	Метаданные маршрута объект-контейнер
5.12.	result.statusName	string	1..1	Название текущего статуса заявки

Пример ответа

```
{
  "result": {
    "metadata": {
      "id": "543c93d4-cdf1-45d2-b71c-b9283b09417a",
      "processId": "9da43288-1169-4c13-88dc-323826b3167f",
      "metadata": {
        ...
      },
      "scopedMetadata": {
        ...
      }
    },
    "statusName": "Консультация завершена без заключения",
    "id": "9da43288-1169-4c13-88dc-323826b3167f",
    "name": "Новая заявка 06.02.2025 12:58:14",
    "workflowId": "f24be524-fb0a-42c3-a108-bcee78081792",
    "metadataId": "543c93d4-cdf1-45d2-b71c-b9283b09417a",
    "currentStageId": "3e897278-3ff0-4d4b-b5a7-4be86d5150bb",
    "currentStageName": "Консультация завершена без заключения",
    "created": "2025-02-06T13:38:22.207803+03:00",
    "updated": "2025-02-06T15:45:53.700881+03:00",
    "businessStatus": null,
    "humanFriendlyId": "TMC02257475328"
  },
  "success": true,
  "errorCode": 0,
  "message": null,
  "stackTrace": null
}
```

4.10. Организация обмена файлами. Компонент XDS

Компонент XDS предназначен для распределенного хранения данных, поступающих в функциональные сервисы платформы N3.Health. 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 метаданных документа из XDS.
- Получение документа из XDS.

4.10.1. Передача файла вложения заявки (POST /api/Xds/xds)

Для передачи объекта файла вложения, прикрепленного к заявке (регистрация данных) в сервисе XDS используется метод POST `BaseUrl/tm-widgets/api/Xds/xds`, позволяющий загрузить файл и в результате получить идентификатор загруженного файла в ответ. Метод предназначен для отправки файлов, прикрепленных к заявке (например, результатов выполненных исследований).

Пример запроса

Тело запроса передается в формате `multipart/form-data`. Пример приведен в форме `curl`-запроса.

```
POST BaseUrl/tm-widgets/api/Xds/xds
accept: text/plain
Content-Type: multipart/form-data
Authorization: <userSessionId>

formFile=@"/C:/Files/testFile.xml"
type="text/xml"
```

Описание параметров ответа:

№	Название	Тип	Крат.	Описание
1.	success	boolean	1..1	Успешность выполнения запроса
2.	errorCode	integer	1..1	Код ошибки 0 - без ошибок
3.	message	string	0..1	Описание ошибки для пользователя null - без ошибок
4.	stackTrace	string	0..1	Техническое описание ошибки null - без ошибок
5.	result	object	0..1	объект-контейнер не указывается при неуспешном вызове
5.1.	result.id	string	1..1	Идентификатор файла в системе хранения
5.2.	result.url	string	1..1	Ссылка для скачивания файла

Пример ответа:

```
{
  "result": {
    "id": "c1785612-88f7-4b1e-a3e9-57c4f4803c54",
    "url": "BaseUrl/tm-widgets/api/Xds/xds?fileId=c1785612-88f7-4b1e-a3e9-57c4f4803c54"
  },
  "success": true,
  "errorCode": 0,
  "message": null,
  "stackTrace": null
}
```

4.10.2. Получение метаданных файла вложения заявки (GET /api/Xds/{id}/info)

Пример запроса

GET **BaseUrl/tm-widgets/api/Xds/<fileId>/info**
Authorization: <userSessionId>

Описание параметров ответа:

№	Название	Тип	Крат.	Описание
1.	result	object	0..1	объект-контейнер (не указывается при неуспешном вызове)
1.1.	result.id	string	1..1	Идентификатор файла в системе хранения
1.2.	result.mimeType	string	1..1	Медиа тип файла
1.3.	result.title	string	1..1	Название файла
1.4.	result.description	null	0..1	Описание файла
1.5.	result.metadata	array	1..1	массив-контейнер прочих метаданных
1.5.1.	(storage_type) result.metadata[]	object	..1	Тип хранения объект-контейнер
1.5.1.1.	(storage_type) result.metadata[].key	string	1..1	Название свойства <i>Допустимые значения</i> <i>storage_type</i>
1.5.1.2.	(storage_type) result.metadata[].value	string	1..1	Тип хранения <i>Допустимые значения</i> <i>Xds</i>
1.5.2.	(extension) result.metadata[]	object	1..1	Расширение файла объект-контейнер
1.5.2.1.	(extension) result.metadata[].key	string	1..1	Название свойства <i>Допустимые значения</i> <i>extension</i>
1.5.2.2.	(extension) result.metadata[].value	string	1..1	Расширение файла <i>Пример</i> <i>pdf</i>
1.5.3.	(is_available) result.metadata[]	object	1..1	Доступность объект-контейнер

№	Название	Тип	Крат.	Описание
1.5.3.1.	(is_available) result.metadata[].key	string	1..1	Название свойства <i>Допустимые значения</i> is_available
1.5.3.2.	(is_available) result.metadata[].value	string	1..1	Доступность <i>Допустимые значения</i> true false
1.5.4.	(size) result.metadata[]	object	1..1	Размер файла объект-контейнер
1.5.4.1.	(size) result.metadata[].key	string	1..1	Название свойства <i>Допустимые значения</i> size
1.5.4.2.	(size) result.metadata[].value	string	1..1	Размер файла в килобайтах <i>Пример</i> 9
1.5.5.	(hash) result.metadata[]	object	1..1	Хэш объект-контейнер
1.5.5.1.	(hash) result.metadata[].key	string	1..1	Название свойства <i>Допустимые значения</i> hash
1.5.5.2.	(hash) result.metadata[].value	string	1..1	Хэш
2.	success	boolean	1..1	Успешность выполнения запроса
3.	errorCode	integer	1..1	Код ошибки <i>0 - без ошибок</i>
4.	message	null	0..1	Описание ошибки для пользователя <i>null - без ошибок</i>
5.	stackTrace	null	0..1	Техническое описание ошибки <i>null - без ошибок</i>

Пример ответа

```
{
  "result": {
    "id": "c1785612-88f7-4b1e-a3e9-57c4f4803c54",
    "mimeType": "application/xml*.xml",
    "title": "testFile",
    "description": null,
    "metadata": [
      {
        "key": "storage_type",
        "value": "Xds"
      },
      {
        "key": "extension",
        "value": "xml"
      },
      {
        "key": "is_available",
        "value": "true"
      },
      {
        "key": "size",
        "value": "4352"
      },
      {
        "key": "hash",
```



```

        "value": "Y5ePDQMJAq6o1t/wDjYm+A=="
    }
  ]
},
"success": true,
"errorCode": 0,
"message": null,
"stackTrace": null
}

```

4.10.3. Получение файла вложения (GET /api/Xds/xds)

Для получения объекта файла вложения, может использоваться метод GET `BaseUrl/tm-widgets/api/Xds/xds`. Метод предназначен для получения файлов по идентификатору.

Пример запроса

```

GET BaseUrl/tm-widgets/api/Xds/xds?fileId=<fileId>
Authorization: <userSessionId>

```

Метод возвращает файл.

5. ОРГАНИЗАЦИЯ ПОЛУЧЕНИЯ ЧАТА ВКС

В процессе движения заявки по определенным маршрутам (например «Врач-пациент», «Врач-врач») для нее генерируются чат и комната ВКС. Подключение к комнате с встроенным в нее чатом возможно по ссылке сервиса MILA N3.Health. Информация о переходах, на которых формируется (после которых может быть доступна) ссылка, указана в Спецификации бизнес-процессов обработки заявок конкретного маршрута.

В типовом решении ссылка может быть получена методом GET `ProcessContext` по пути `result.externalRoom.URL`:

```

{
  "result": {
    ...
    "externalRoom": {
      "URL": "BaseUrl/mila_vcs/<Идентификатор комнаты>"
    },
    ...
  },
  "success": true,
  "errorCode": 0,
  "message": null,
  "stackTrace": null
}

```

Для авторизации в ВКС комнате требуется дополнить ссылку параметром в виде идентификатора сессии пользователя (`userSessionId`):

```

BaseUrl/mila_vcs/<Идентификатор комнаты>?sessionId=<userSessionId>

```

6. 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