



**@СТРАЛ**  
www.astral.ru

ООО «АСТРАЛ-СОФТ»

**Руководство пользователя ИС ЭПД  
Описание API информационной системы  
электронных перевозочных документов**

Калуга

## СОДЕРЖАНИЕ

<b>1. ОПИСАНИЕ.....</b>	<b>5</b>
1.1. ФУНКЦИОНАЛЬНОСТЬ СЕРВИСА.....	5
1.2. ПРЕИМУЩЕСТВА ТРАНСПОРТНОГО ЭДО .....	5
1.3. ПЕРЕД НАЧАЛОМ РАБОТЫ.....	5
1.4. СПИСОК ВОЗМОЖНЫХ ОШИБОК СЕРВИСА .....	7
<b>2. СЦЕНАРИИ РАБОТЫ .....</b>	<b>10</b>
2.1. ОТПРАВКА УЧАСТНИКАМ ПЕРЕВОЗКИ ДОКУМЕНТА О ПЕРЕВОЗКЕ.....	10
2.2. ПОДТВЕРЖДЕНИЕ ПРИЕМА ВХОДЯЩЕГО ДОКУМЕНТА ОТ УЧАСТНИКОВ ПЕРЕВОЗКИ .....	10
2.3. ЗАПРОС НА ФОРМИРОВАНИЕ СЛУЖЕБНОГО ДОКУМЕНТА .....	10
2.4. ПОЛУЧЕНИЕ СТАТУСА ДОКУМЕНТООБОРОТА .....	11
2.5. ПОЛУЧЕНИЕ СТАТУСА ДОКУМЕНТА В ДОКУМЕНТООБОРОТЕ.....	11
2.6. ПОЛУЧЕНИЕ НОВОГО ПУЛА УИДОВ ДЛЯ ДОКУМЕНТОВ Т1ЭТРН .....	11
2.7. ПОЛУЧЕНИЕ СПИСКА ДОСТУПНЫХ К ИСПОЛЬЗОВАНИЮ УИДОВ ИЗ РАНЕЕ НАЗНАЧЕННЫХ ДЛЯ ДОКУМЕНТОВ Т1ЭТРН .....	12
2.8. УВЕДОМЛЕНИЕ О НАЗНАЧЕНИИ УИДА ДЛЯ РАНЕЕ ОТПРАВЛЕННОГО ДОКУМЕНТА Т1ЭТРН БЕЗ УКАЗАНИЯ УИДА 12	
2.9. ЗАПРОС АРХИВА ДОКУМЕНТОВ ПО ДОКУМЕНТООБОРОТУ .....	12
<b>3. МЕТОДЫ.....</b>	<b>13</b>
3.1. РАБОТЫ С СОБЫТИЯМИ .....	13
3.1.1. <i>POST /events</i> .....	13
Публикация событий для сервисов АСТРАЛ ПЛАТФОРМЫ.....	13
3.1.2. <i>GET /events</i> .....	14
3.1.3. <i>GET /events/{event_uuid}</i> .....	15
Метод получения конкретного события по идентификатору.....	15
3.1.4. <i>POST /events/confirm</i> .....	16
Метод подтверждения получения событий. ....	16
3.1.5. <i>GET /events/unconfirmed</i> .....	16
Метод получения списка неподтвержденных событий. ....	16
3.1.6. <i>GET /events/{event_uuid}{ext}</i> .....	17
3.2. РАБОТА С ФАЙЛАМИ .....	17
3.2.1. <i>POST /files</i> .....	17
3.2.2. <i>GET /files/{file_uuid}</i> .....	18
3.2.3. <i>GET /files/{file_uuid}.json</i> .....	18
3.2.4. <i>PATCH /files/{file_uuid}</i> .....	19
3.2.5. <i>DELETE /files/{file_uuid}</i> .....	19
<b>4. API ИНТЕГРАЦИОННОЙ ШИНЫ V1.0. ....</b>	<b>20</b>
4.1. ВЕБСОКЕТЫ /WS .....	20
4.2. РАБОТА С ФАЙЛАМИ .....	20
4.2.1. <i>Сохранение файла /save_file</i> .....	20
4.2.2. <i>Получение файла /get_file/{id}</i> .....	20
4.3. АВТОРИЗАЦИЯ /AUTH .....	21
4.3.1. <i>Вход (получение пары jwt) /sign_in</i> .....	21
4.3.2. <i>Обновление Access Token/refresh_tokens</i> .....	22
<b>5. КОНВЕРТ И СООБЩЕНИЯ .....</b>	<b>23</b>
5.1. ОБЩЕЕ ОПИСАНИЕ .....	23

5.1.1.	Конверт.....	23
5.1.2.	Сообщение .....	23
5.2.	ПРИМЕРЫ СООБЩЕНИЙ В КОНВЕРТЕ .....	23
5.2.1.	Исходящее сообщение.....	23
5.2.2.	Входящее сообщение .....	24
5.2.3.	Входящее сообщение об ошибке .....	25
<b>6.</b>	<b>СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ.....</b>	<b>26</b>
6.1.	Документы .....	26
6.1.1.	Процесс создания служебного документа.....	26
6.1.2.	Процесс обработки документа "Информация от грузоотправителя (Т1 ЭТрН)" без УИД_ТрН.....	27
6.1.3.	Процесс обработки документа с заполненным УИД_ТрН.....	28
6.1.4.	Процесс получения статуса документа.....	29
6.2.	Уиды ГИС ЭПД.....	29
6.2.1.	Процесс запроса пула новых уидов.....	29
6.2.2.	Процесс запроса пула уидов, которые ранее были выданы.....	30
6.3.	Документооборот .....	30
6.3.1.	Процесс обработки изменения состояния ДО .....	30
6.3.2.	Процесс получения статуса ДО.....	30
6.3.3.	Процесс получения списка всех документов ДО.....	31
6.4.	REGISTRATION.....	31
6.4.1.	Регистрация.....	31
6.4.2.	Создание абонента .....	31
6.4.3.	Обновление абонента.....	32
6.4.4.	Создание сотрудника абонента .....	32
6.4.5.	Обновление сотрудника абонента .....	32
6.4.6.	Добавление сертификата сотрудника организации .....	33
6.4.7.	Получение данных абонента .....	33
6.5.	Приглашения .....	33
6.5.1.	Отправить приглашение и принять приглашение.....	34
6.5.2.	Отправить приглашение и отклонить приглашение .....	34
6.5.3.	Получить статус приглашений .....	35
6.5.4.	Получить входящие приглашения.....	35
6.5.5.	Получить ИД ЭДО абонента.....	35
<b>7.</b>	<b>МОДЕЛИ СООБЩЕНИЙ.....</b>	<b>36</b>
7.1.	Документы .....	36
7.1.1.	Командные модели .....	36
7.1.2.	Ответные модели .....	39
7.1.3.	Дополнительные модели .....	42
7.2.	Документооборот .....	47
7.2.1.	Командные модели .....	47
7.2.2.	Ответные модели .....	48
7.2.3.	Дополнительные модели .....	50
7.3.	УИДы.....	53
7.3.1.	Командные модели .....	53
7.3.2.	Ответные модели .....	54
7.3.3.	Дополнительные модели .....	56
7.4.	Ошибки .....	56
7.4.1.	Ответные модели .....	56

7.5.	REGISTRATION .....	57
7.5.1.	Описание командных моделей .....	57
7.5.2.	Описание ответных моделей .....	65
7.5.3.	Описание дополнительных моделей .....	69
7.6.	РАБОТА С ПРИГЛАШЕНИЯМИ .....	76
7.6.1.	Командные модели .....	77
7.6.2.	Ответные модели .....	81
7.6.3.	Дополнительные модели .....	82
<b>8.</b>	<b>ПРИЛОЖЕНИЕ ДЛЯ ВОДИТЕЛЕЙ.....</b>	<b>84</b>
8.1.	НАСТРОЙКА .....	84
8.2.	АВТОРИЗАЦИЯ .....	85
8.3.	ПРОХОЖДЕНИЕ ПРЕДРЕЙСОВОГО МЕДИЦИНСКОГО ОСМОТРА (ПОДПИСЬ ВОДИТЕЛЯ) .....	85
8.4.	ПРИЕМ ТРАНСПОРТНОГО СРЕДСТВА (ПОДПИСЬ ВОДИТЕЛЯ).....	88
8.5.	ВНЕСЕНИЕ ПОКАЗАНИЙ ОДОМЕТРА ПРИ ВЫЕЗДЕ/ЗАЕЗДЕ.....	90
8.6.	ВНЕСЕНИЕ ПОКАЗАНИЙ ОСТАТКА ГСМ ПРИ ВЫЕЗДЕ/ЗАЕЗДЕ.....	91
8.7.	QR - КОД ПУТЕВОГО ЛИСТА .....	92
8.8.	ВНЕСЕНИЕ ПОКАЗАНИЙ ПО КРУГОРЕЙСАМ .....	94
8.9.	ВНЕСЕНИЕ ПОКАЗАНИЙ ПО ЗАПРАВКАМ В ТЕЧЕНИЕ РЕЙСА .....	95
8.10.	ПОСЛЕРЕЙСОВЫЙ МЕДИЦИНСКИЙ ОСМОТР .....	97

## 1. Описание

### 1.1. Функциональность сервиса.

- Обмен перевозочными электронными документами (накладные, заказ-наряды, сопроводительные ведомости)
- Поддержка всех требуемых ФНС форматов ЭПД, а также обмен неформализованными документами
- Обеспечение подписания ЭПД всеми участниками: грузоотправители, грузополучатели, грузоперевозчики, исполнители, кладовщики, продавцы-кассиры, водители
- Визуализация всех типов документов
- Мгновенное уведомление участников перевозки о деталях приемки-передачи груза
- Интеграция с 1С, SAP и другими информационными системами

### 1.2. Преимущества транспортного ЭДО

- Скорость документооборота, повышение эффективности информационного и документационного обеспечения
- Сокращение операционного цикла и увеличение оборачиваемости. Транспортная компания получает оплату за услуги в среднем на 20 дней быстрее
- Контроль движения документов на всех этапах жизненного цикла, снижение рисков потери документов и ошибок в процессах
- Снижение стоимости операций и экономия на обработке и доставке документов
- Возможность организовать дополнительные сервисы, например: доступ к финансированию, страхованию и другие

### 1.3. Перед началом работы

Перед началом работы с сервисом ЭПД, рекомендуется ознакомиться со следующими разделами документации:

#### 1. Концепция работы с файлами

**Файл** - это именованный набор произвольных данных. Система не накладывает ограничений на тип или имя файла (если это не противоречит логике обработки событий, в которых участвует файл), но ограничивает максимально возможный размер - 128 Мб.

**Список контроля доступа (ACL)** - это набор правил, определяющих соответствие участников системы и допустимых для них операций с файлами. Каждый файл обладает этим списком, что делает возможным использование механизмов наследования и переопределения.

#### 2. Концепция работы с событиями

Астрал.Платформа построена по принципу асинхронной ESB-системы, а взаимодействие осуществляется с помощью обмена событиями.

**Участник системы** - это субъект взаимодействия (партнерская система, сервис компании или другой поставщик услуг), интегрированный в систему и обладающий уникальным в рамках системы идентификатором.

**Событие** - это набор данных установленного формата, передаваемый системой между участниками и, возможно, связанный с другими событиями. События также могут содержать в себе идентификаторы файлов.

**Код события** - это уникальный набор символов, идентифицирующий тип данных и способ их обработки участниками.

**Отправитель (Producer)** - идентификатор участника системы, который отправил новое событие. Генерируется автоматически при отправке, на основании идентифицирующего заголовка X-API-Key.

**Получатель (Consumer)** - идентификатор участника системы, который должен получить данное событие. Должен быть указан отправителем. Может быть указано несколько получателей для одного события, в этом случае его получат все участники, перечисленные в списке.

Событие может иметь ссылку на **родительское событие (Parent)**. Это требуется для выстраивания цепочки этапов процесса. Например, исходное событие запроса на проверку подписи файла не имеет родительского, но будет являться родительским для дочернего события результата проверки.

В широком смысле, Астрал.Платформа - это особый менеджер для работы с очередями, реализующий **паттерн pub/sub**. Это определяет логику использования API. События, аналогично задачам в очереди, могут быть опубликованы их производителем по одному или группой. А потребитель создает одну или несколько подписок для получения списка новых задач.

Платформа предоставляет **long-polling API** по работе с событиями. Этот способ обеспечения асинхронного взаимодействия сочетает простоту использования и высокую скорость доставки сообщений в системе. Поскольку каждый новый запрос к API требует накладных расходов на сетевое взаимодействие, для повышения эффективности утилизации ресурсов рекомендуется использовать максимально длительный интервал ожидания, обеспечивающий стабильность соединения, чаще всего в интервале от 60 до 600 сек.

### 3. Концепция обработки ошибок

**Ошибка** - это объект, сигнализирующий о сбое в процессе обработки запроса или события. Он содержит информацию о моменте и причине возникновения, коды для автоматизации обработки, ссылку на документацию и, возможно, другие необходимые сведения. Ошибки делятся на две группы: клиентская и серверная.

**Ошибка клиента** означает, что прерывание обработки было прямо или косвенно вызвано конечным пользователем или действиями системы, которая интегрируется с Платформой. К таким ошибкам можно отнести передачу некорректных файлов, параметров, отказ пользователя подтвердить операцию и т.п. Исправление клиентских ошибок частично или полностью зависит от пользователя и/или разработчиков интегрируемой системы.

**Ошибкой сервера** считается любая другая ошибка, поэтому для их исправления следует обратиться в техническую поддержку.

Все ошибки возвращаются в формате [RFC 7807](#) с дополнительными полями:

- **Id string** - уникальный идентификатор ошибки (UUIDv4), должен использоваться при взаимодействии с технической поддержкой для точной локализации проблемы
- **code string** - код типа ошибки, может использоваться для идентификации причины и запуска особого обработчика на стороне клиента
- **type string** - тип ошибки, согласно RFC это абсолютная ссылка на ресурс с документацией. В случае отсутствия ссылки, допускается about:blank
- **title string** - краткое описание причины ошибки, не зависит от данных и не изменяется между ошибками с одинаковым кодом
- **status integer** - согласно RFC это код HTTP-статуса ошибки, может использоваться для идентификации причины и запуска особого обработчика на стороне клиента, но поле code дает более точную классификацию
- **created string** - момент возникновения в формате RFC 3339 с дробной частью (с точностью до нс)
- **detail string** - необязательное полное описание причины ошибки, зависит от данных и содержит форматированные значения, может использоваться для диагностики или для отображения пользователю

Ошибку можно получить одним из двух способов:

- Базовое API при работе с файлами и событиями возвращает ошибку синхронно, в теле HTTP-ответа.
- Любой из сервисов может вернуть событие client.error или server.error и передать ошибку в нем

В случае ошибок в событиях, в поле parent будет указан идентификатор события, которое вызвало ошибку. Код события для клиентской и серверной ошибок отличается, чтобы можно было сделать маршрутизацию на стадии разбора события.

Все коды ошибок по всем сервисам, а также их краткое описание и причины возникновения перечислены в [отдельном разделе](#).

Пример содержимого ответа или события:

```
{
  "id": "bc245b1887e4405782f26af8f3f37981",
  "code": "reg4049",
  "type":
  "https://demo.platform.astral.ru/documentation?tab=4&dataIndex=3#reg4049",
  "title": "Нет подходящих данных",
  "status": 404,
  "created": "2020-03-28T08:01:47.914775677Z",
  "detail": "Нет абонента с ID сертификата \"e90b02ae-c1ed-4075-bd08-6069b4f5b1d8\""
}
```

#### 1.4. Список возможных ошибок сервиса

##### #epd4001

Некорректное значение параметра

В epd.uuid.check.request

Максимальное значение параметра Количество не должно превышать 1000

#### **#epd40011**

Не указан обязательный параметр

В epd.service.doc.request

При запросе УОУ, параметр Комментарий является обязательным

#### **#epd40021**

Необходимо указать родительское событие

В epd.doc.accept, epd.service.doc.request

Не указан параметр parent

#### **#epd40022**

Неверный код родительского события

В epd.doc.accept, epd.service.doc.request

Родительское событие должно быть epd.incoming

#### **#epd4003**

Документ не подписан

В epd.doc.send

Исходящий документ не содержит файл подписи - параметр ФайлПодписи или не указан запрос подписи через приложение DSS - структура ПараметрыПодписиDSS

#### **#epd40041**

Срок действия сертификата истек

В epd.doc.send

Срок действия сертификата с идентификатором '%s' истек

#### **#epd4041**

Сертификат не существует

В epd.doc.send, epd.service.doc.request

Сертификат с идентификатором '%s' не существует

#### **#epd4061**

## Операция подписания отклонена

В epd.doc.send

Пользователь отклонил запрос подписания документа в приложении DSS

Превышено время ожидания подтверждения

### **#epd5009**

Ошибка обработки запроса

Возможна во всех событиях сервиса

Временные неполадки на стороне сервера

При регулярном воспроизведении, обратитесь в техническую поддержку

## 2. Сценарии работы

### 2.1. Отправка участникам перевозки документа о перевозке

1. В файловое хранилище загружается документ, который требуется отправить, и файл подписи к документу. В ответ возвращаются метаданные по загруженным файлам.  
Метод отправки запроса: **POST/files**
2. После загрузки документа и файла подписи к документу, в сервис ЭПД (код сервиса - "epd") необходимо опубликовать событие по отправке документа (код события - "epd.doc.send", при отправке документа в ответ на входящий документ - необходимо указать родительское событие "epd.incoming"). В ответ возвращаются метаданные по событию.  
Метод отправки запроса: **POST/events**
3. \*Опционально. При подписании исходящего документа через DSS\* - сервис ЭПД публикует уведомление (код события, в случае успешного подписания - "epd.doc.signed"; в случае ошибки - "client.error" либо "server.error").  
Метод отправки запроса: **GET/events**
4. После обработки события по отправке документа, сервис ЭПД публикует событие с информацией об исходящем документе в рамках документооборота (код события, в случае успешной отправки документа - "epd.docflow"; в случае ошибки - "client.error" либо "server.error").  
Метод отправки запроса: **GET/events**

### 2.2. Подтверждение приема входящего документа от участников перевозки

1. При поступлении ответного документа на ранее отправленный документ, сервис ЭПД публикует событие о входящем документе (код события - "epd.incoming") с метаданными о входящем документе.  
Метод отправки запроса: **GET/events**
2. После получения события с информацией о входящем документе, в сервис ЭПД (код сервиса - "epd") необходимо опубликовать событие о подтверждении приема документа (код события - "epd.doc.accept" с указанием родительского события "epd.incoming" из пункта 1).  
Метод отправки запроса: **POST/events**
3. При необходимости выгрузки входящего документа из файлового хранилища следует выполнить запрос на получение файла.  
Метод отправки запроса: **GET/files/{file\_uuid}**

### 2.3. Запрос на формирование служебного документа

1. Для формирования служебного документа на входящий документ в цепочке документооборота по перевозке, в сервис ЭПД (код сервиса - "epd") публикуется событие с запросом служебного документа (код события - "epd.service.doc.request", с указанием родительского события "epd.incoming" в котором был получен входящий документ). В ответ возвращаются метаданные по событию.  
Метод отправки запроса: **POST/events**

2. После обработки запроса, сервис ЭПД публикует событие с информацией о сформированном служебном документе (код события, в случае успешного формирования документа - **"epd.service.doc.response"**; в случае ошибки - **"client.error"** либо **"server.error"**).

Метод отправки запроса: **GET/events**

3. Для выгрузки файла служебного документа из файлового хранилища следует выполнить запрос на получение файла.

Метод отправки запроса: **GET/files/{file\_uuid}**

#### 2.4. Получение статуса документооборота

По умолчанию при любом изменении статуса документооборота сервис ЭПД публикует событие **"epd.docflow.status.info"**.

С целью уточнения статуса документооборота по запросу, необходимо:

1. В сервис ЭПД (код сервиса - **"epd"**) опубликовать событие с запросом статуса документооборота (код события - **"epd.docflow.status.request"**). В ответ возвращаются метаданные по событию.

Метод отправки запроса: **POST/events**

2. После обработки события с запросом, сервис ЭПД публикует новое событие о статусе документооборота (код события, в случае успешного выполнения запроса - **"epd.docflow.status.info"**; в случае ошибки - **"client.error"** либо **"server.error"**).

Метод отправки запроса: **GET/events**

#### 2.5. Получение статуса документа в документообороте

По умолчанию при любом изменении статуса документа в документообороте сервис ЭПД публикует событие **"epd.doc.status.info"**.

С целью уточнения статуса документа в документообороте по запросу, необходимо:

1. В сервис ЭПД (код сервиса - **"epd"**) опубликовать событие с запросом статуса документа в документообороте (код события - **"epd.doc.status.request"**). В ответ возвращаются метаданные по событию.

Метод отправки запроса: **POST/events**

2. После обработки события с запросом, сервис ЭПД публикует событие о статусе документа (код события, в случае успешного выполнения запроса - **"epd.doc.status.info"**; в случае ошибки - **"client.error"** либо **"server.error"**).

Метод отправки запроса: **GET/events**

#### 2.6. Получение нового пула УИДов для документов Т1ЭТрН

1. Для получения нового пула УИДов для документа Т1ЭТрН в сервис ЭПД (код сервиса - **"epd"**) направляется событие с запросом назначения нового пула УИДов (код события - **"epd.uuid.new.request"**). В ответ на запрос возвращаются метаданные по событию.

Метод отправки запроса: **POST/events**

2. После обработки запроса, сервис ЭПД публикует событие с результатом назначения УИДов, представленных в виде JSON-объекта, помещенного в файл (код события, в случае успешного назначения - **"epd.uuid.new.response"**);

в случае ошибки - "**client.error**" либо "**server.error**").

Метод отправки запроса: **GET/events**

3. Для выгрузки файла с назначенными УИДами из файлового хранилища следует выполнить запрос на получение файла.  
Метод отправки запроса: **GET/files/{file\_uuid}**

## 2.7. Получение списка доступных к использованию УИДов из ранее назначенных для документов Т1ЭТрН

1. Для получения списка доступных к использованию УИДов из ранее назначенных для документов Т1ЭТрН, в сервис ЭПД (код сервиса - "**epd**") направляется событие с запросом на получение списка доступных к использованию УИДов (код события - "**epd.uuid.check.request**"). В ответ на запрос возвращаются метаданные по событию.  
Метод отправки запроса: **POST/events**
2. После обработки запроса, сервис ЭПД публикует событие с результатом выгрузки доступных УИДов, представленных в виде JSON-объекта, помещенного в файл (код события, в случае успешного выполнения - "**epd.uuid.check.response**"; в случае ошибки - "**client.error**" либо "**server.error**").  
Метод отправки запроса: **GET/events**
3. Для выгрузки файла со списком доступных к использованию УИДов из файлового хранилища следует выполнить запрос на получение файла.  
Метод отправки запроса: **GET/files/{file\_uuid}**

## 2.8. Уведомление о назначении УИДа для ранее отправленного документа Т1ЭТрН без указания УИДа

В случае, когда первичный документ Т1ЭТрН цепочки документооборота по перевозке был отправлен без указания УИД, сервис ЭПД опубликует событие с уведомлением о назначении УИДа на цепочку документооборота, код события - "**epd.uuid.defined**".

Метод отправки запроса: **GET/events**

## 2.9. Запрос архива документов по документообороту

1. Для получения архива документов по перевозке в сервис ЭПД (идентификатор сервиса - "**epd**") направляется запрос с идентификатором документооборота (код события - "**epd.docflow.archive.request**"). В ответ на запрос возвращаются метаданные по событию.  
Метод отправки запроса: **POST/events**
2. После обработки запроса, сервис ЭПД публикует событие с результатом выгрузки архива документов (код события, в случае успешного выполнения - "**epd.docflow.archive.response**"; в случае ошибки - "**client.error**" либо "**server.error**").  
Метод отправки запроса: **GET/events**
3. Для выгрузки архива документооборота из файлового хранилища следует выполнить запрос на получение файла.  
Метод отправки запроса: **GET/files/{file\_uuid}**

### 3. Методы

#### 3.1. Работы с событиями

##### 3.1.1. POST /events

Публикация событий для сервисов Астрал Платформы

Опубликовать можно одно или несколько событий одновременно. У них могут быть различные коды, получатели и родительские события. При публикации списка, **каждое событие публикуется индивидуально**, поэтому, в случае ошибки формирования одного из событий, другие всё равно будут опубликованы и система продолжит работу. Система гарантирует, что события будут поступать получателю в порядке их публикации, но не гарантирует, что получатель обработает их в этом порядке.

**Входным параметром** является объект JSON, общая структура которого имеет вид:

```
{
  "events": [ список публикуемых событий
    {
      "code": string - код типа события
      "parent": string - необязательный идентификатор родительского события
      "consumer": [string] - список сервисов для обработки события
      "data": { } - параметры для обработки события в сервисе
    }
  ]
}
```

**Выходным параметром** является объект JSON, содержащий следующую структуру:

```
{
  "status": integer {100-600} - статус обработки запроса,
  "type": string uri - ссылка на документацию по ошибке, about:blank, если ошибки нет,
  "title": string - краткое описание результата запроса,
  "events": [
    {
      "id": string string [a-f0-9]{32} - идентификатор события,
      "errors": [string] - список ошибок возникших при публикации,
      "code": string - код публикуемого события,
      "producer": string - идентификатор отправителя события,
      "consumer": [string] - список сервисов для обработки события,
      "created": string rfc3339 - дата формирования события,
      "data": { } - параметры для обработки события в сервисе
    }
  ]
}
```

## Обработка

- Поля `id` и `errors` в ответе взаимоисключающие.
- Если в информации по событию в ответе есть поле `id`, то это событие успешно опубликовалось.

### 3.1.2. GET /events

Получение входящих событий от сервисов Астрал.Платформы

#### Параметры:

- **timeout** *integer* - максимальное время ожидания новых событий, в секундах
  - Не менее 10 сек
  - Не более 600 сек
  - По-умолчанию 60 сек
  - Чем больше значение, тем меньше накладных расходов на соединение
  - Следует указывать настолько большим, насколько возможно по конфигурации сети
  - Некорректные значения вызывают ошибку [api40011](#)
- **limit** *integer* - максимальное кол-во событий, которые необходимо получить
  - Не менее 1
  - Не более 100
  - По-умолчанию 100
  - Некорректные значения вызывают ошибку [api40012](#)

#### Обработка:

- Работает по технологии **long polling**
- При наличии новых событий, ответ отдается сразу же
- При отсутствии, сервер не закрывает запрос в течение времени таймаута
- Если события поступили до этого момента, ответ возвращается сразу же
- Может вызываться параллельно, результаты не пересекаются, каждый запрос получит свои события
- Каждое новое событие выдается ровно один раз, после чего оно считается прочитанным

#### Рекомендации:

- Нужно использовать в "вечном" цикле, каждый раз выполняя новый запрос сразу же после завершения предыдущего
- В минимальном варианте достаточно одного ожидающего запроса. Он будет получать все входящие события
- Успешную обработку события необходимо подтвердить с помощью `POST /events/confirm`
- Повторно получить неподтвержденные события можно с помощью `GET /events/unconfirmed`

#### Ответ:

```
{
```

```

"status": integer {100-600} - статус обработки запроса,
"type": string uri - ссылка на документацию по ошибке, about:blank, если
ошибки нет,
"title": string - краткое описание результата запроса,
"events": [
  {
    "id": string string [a-f0-9]{32} - идентификатор события,
    "code": string - код публикуемого события,
    "producer": string - идентификатор отправителя события,
    "consumer": [string] - список сервисов для обработки события,
    "created": string rfc3339 - дата формирования события,
    "data": { } - параметры для обработки события в сервисе
  }
]
}

```

### 3.1.3. GET /events/{event\_uuid}

Метод получения конкретного события по идентификатору

Может быть полезен для уточнения данных события, повторной обработки, проверки корректности публикации. Текущий статус события не влияют на работу метода: данные можно получить в любой момент, как до его обработки, так и после подтверждения.

**Входным параметром** является идентификатор события - *string [a-f0-9]{32}* **event\_uuid**

**Выходным параметром** по получению результата является объект JSON, содержащий следующую структуру:

```

{
  "status": integer{100-600} - статус обработки запроса,
  "type": string uri - ссылка на документацию по ошибке, about:blank, если
ошибки нет,
  "title": string - краткое описание результата запроса,
  "events": [
    {
      "id": string [a-f0-9]{32} - идентификатор события,
      "code": string - код публикуемого события,
      "producer": идентификатор отправителя события,
      "parent": string [a-f0-9]{32} - идентификатор родительского
обработанного события,
      "consumer": [string] - список сервисов для обработки события,
      "created": string rfc3339 - дата формирования события,
      "data": { } - результат выполненного события
    }
  ]
}

```

### 3.1.4. POST /events/confirm

Метод подтверждения получения событий.

Отправка идентификатора события в этот метод сигнализирует системе о том, что событие успешно получено и обработано, получатель несет полную ответственность за дальнейшую судьбу данных, а событие пропадает из списка GET /events/unconfirmed. Тем не менее, при необходимости, можно получить данные события по идентификатору GET /events/{event\_uuid}.

**Входным параметром** является объект JSON, который имеет следующую структуру:

```
{
  "events_id": [ array - список идентификаторов событий
    string [a-f0-9]{32}
  ]
}
```

**Выходным параметром** по получению результата является объект JSON, содержащий следующую структуру:

```
{
  "status": integer {100-600} - статус обработки запроса,
  "type": string uri - ссылка на документацию по ошибки, about:blank, если
  ошибки нет,
  "title": string - краткое описание результата запроса,
  "count": integer - количество подтвержденных событий
}
```

### 3.1.5. GET /events/unconfirmed

Метод получения списка неподтвержденных событий.

Эта возможность - один из инструментов обеспечения **гарантии доставки** и надежности системы. Если принимающая сторона по какой-то причине не смогла обработать и подтвердить часть событий, их всегда можно снова получить с помощью данного метода (неограниченное кол-во раз, вплоть до подтверждения). При этом "застрявшие" события не мешают обработке основного потока новых событий в GET /events, не допуская деградации системы для пользователя.

**Входным параметром** является:

- **limit integer** - максимальное количество событий, которое процесс готов получить;
- по умолчанию **100**, максимально допустимое количество **100**

**Выходным параметром** является объект JSON, содержащий следующую структуру:

```
{
  "status": integer {100-600} - статус обработки запроса,
  "type": string uri - ссылка на документацию по ошибки, about:blank, если
  ошибки нет,
```

```

"title": string - краткое описание результата запроса,
"events": [
  {
    "id": string string [a-f0-9]{32} - идентификатор события,
    "code": string - код события,
    "producer": string - идентификатор отправителя (сервиса),
    "consumer": [string] - идентификатор получателя,
    "created": string rfc3339 - дата формирования события,
    "data": { } - параметры для обработки события в сервисе
  }
]
}

```

### 3.1.6. GET /events/{event\_uuid}{ext}

Метод скачивания архива с файлами и данными события.

#### Входные параметры:

- **event\_uuid** string [a-f0-9]{32} - идентификатор события
- **ext** string .tar.gz, .zip - расширение архива с файлами из события
- **strict** boolean - генерация ошибки в случае невозможности сборки архива со всеми файлами (например, если они ранее были удалены)

#### Формат ответа:

Архив с файлами события в бинарном виде (имя файла - его идентификатор без расширения). Также в каждом архиве присутствует файл event.json с данными события.

## 3.2. Работа с файлами

### 3.2.1. POST /files

Метод для загрузки файлов в сервис Астрал.Платформа

#### Входные параметры:

- Форма с одним или несколькими файлами. Должна быть сформирована как **multipart/form-data**. Имя поля в форме (параметр name) не имеет рекомендованного значения - обработаны будут все файловые поля.
- Если есть необходимость загрузить файл, кодированный в base64, можно указать файлу заголовок "Content-Transfer-Encoding": "base64" и он будет декодирован в момент загрузки.

#### Формат ответа:

```

{
  "status": integer{100-600} - HTTP-статус обработки запроса,
  "type": string uri - ссылка на документацию по ошибке, about:blank, если
  ошибки нет,
  "title": string - краткое описание результата запроса,

```

```

"detail": string - развернутое описание результата запроса,
"files": [
  {
    "id": string [a-f0-9]{32} - уникальный идентификатор файла, назначается
автоматически
    "name": string - оригинальное имя файла, из поля "filename" в форме
    "size": integer - размер файла (в байтах), после base64-раскодирования,
если оно было
    "mime": string - тип содержимого файла, из заголовка Content-Type файла
    "storage": string - тип хранилища (standard или cold),
    "created": string rfc3339 - дата создания файла, автоматически
выставляется в момент начала загрузки
    "updated": string rfc3339 - дата обновления файла, автоматически
выставляется в момент успешного сохранения
    "acl": {
      "owner": string [a-f0-9]{32} - владелец файла, автоматически
назначается тот, кто загружает
      "access": {}, - карта доступов к использованию файла, автоматически
заполняется для владельца
    }
  }
]
}

```

### 3.2.2. GET /files/{file\_uuid}

Метод скачивания файла

**Входные параметры:**

- **file\_uuid** string [a-f0-9]{32} - идентификатор файла

**Формат ответа:**

Содержимое файла, в бинарном виде, без сжатия и кодирования

### 3.2.3. GET /files/{file\_uuid}.json

Метод получения метаданных по файлу

**Входные параметры:**

- **file\_uuid** string [a-f0-9]{32} - идентификатор файла

**Формат ответа:**

```

{
  "status": integer{100-600} - HTTP-статус обработки запроса,
  "type": string uri - ссылка на документацию по ошибке, about:blank, если
ошибки нет,
  "title": string - краткое описание результата запроса,
  "file": {

```

```

"id": string [a-f0-9]{32} - уникальный идентификатор,
"name": string - оригинальное имя,
"size": integer - размер (в байтах),
"mime": string - тип содержимого,
"storage": string - тип хранилища (standard или cold),
"created": string rfc3339 - дата создания,
"updated": string rfc3339 - дата последнего обновления,
"acl": { - список контроля доступа на модификацию и чтение
  "owner": string - владелец,
  "access": { } - карта доступов
}
}
}

```

### 3.2.4. PATCH /files/{file\_uuid}

Метод модификации файла

**Входные параметры:**

- **file\_uuid** string [a-f0-9]{32} - идентификатор файла
- Форма с JSON-описанием изменяемых параметров
  - Поддерживается только изменение типа хранилища
  - Если файл уже содержится в указанном хранилище, перемещения не происходит

```

{
  "storage": string - тип хранилища, куда следует переместить файл,
  "standard" или "cold"
}

```

**Формат ответа:**

Структура метаданных файла, аналогичная ответу на запрос [GET /files/{file\\_uuid}.json](#)

### 3.2.5. DELETE /files/{file\_uuid}

Метод удаления файла.

**Входные параметры:**

- **file\_uuid** string [a-f0-9]{32} - идентификатор файла

**Формат ответа:**

Пустой ответ с HTTP-статусом 200 говорит о том, что файл либо не существует, либо успешно удален

## 4. API Интеграционной шины v1.0.

### 4.1. Вебсокеты /ws

Работа с веб сокета идет через систему отправки и получения конвертов.

[Обязательно] Header: Authorization: Bearer {token}

Документация к API Интеграционной шины по загрузке и сохранению файла.

base path: /api/v1/

### 4.2. Работа с файлами

#### 4.2.1. Сохранение файла /save\_file

Method: POST

Header: Authorization: Bearer {access\_token}

Content-type: multipart/form-data

Input Object Body **\*required**:

```
{
  "filename": "string",
  "id": "string",
  "type": "string"
}
```

data **\*required** file (formData)

#### Response

Error Code **200**: Файл успешно сохранен

Content-Type: text/plain

Charset: UTF-8

```
// Response Body
"string"
```

Error Code **401**: Неправильные данные авторизации Error Code **500**: Файл не найден / Внутренняя ошибка

#### 4.2.2. Получение файла /get\_file/{id}

Method: GET

Header: Authorization: Bearer {access\_token}

Parameters: id (**\*required**) in `path`

## Response

Error Code **200**: Файл получен

Content-Disposition: attachment; filename=%file% Content-Type: %file\_type%

Error Code **401**: Неправильные данные авторизации Error Code **500**: Файл не найден / Внутренняя ошибка сервера

### 4.3. Авторизация /auth

Требуется для генерации пары jwt, где в дальнейших запросах **Access Token** будет отправляется в заголовке **Authorization** на контрольные точки по работе с файлами.

Время жизни AccessToken:**120 минут (два часа)** \*Время жизни RefreshToken: **30 дней\***

#### 4.3.1. Вход (получение пары jwt) /sign\_in

Method: POST

Content-type: application/json

```
// RequestBody
{
  "username": "string",
  "password": "string"
}
```

**Response** application/json

Error Code **200**: Пользователь успешно авторизован

```
// ResponseBody
{
  "access_token": "string",
  "refresh_token": "string"
}
```

Error Code **400**: Некорректно отправленный запрос / Невалидные данные авторизации Error Code **406**: Пользователя с такими данными не существует Error Code **500**: Произошла внутренняя ошибка

### 4.3.2. Обновление Access Token/refresh\_tokens

Method: POST

Content-type: application/json

```
// RequestBody
{
  "refresh_token": "string"
}
```

#### Response

Error Code **200**: Access Token успешно обновлен

```
// ResponseBody
{
  "access_token": "string"
}
```

Error Code **400**: Некорректно отправлен составленный запрос Error Code **406**: Срок действия Refresh Token истёк Error Code **500**: Произошла внутренняя ошибка

## 5. Конверт и сообщения

Данный раздел содержит описание специального конверта сообщений, необходимого для обмена с системой оператора ЭПД, а также информацию как их собирать.

### 5.1. Общее описание

#### 5.1.1. Конверт

Конверт сообщений - это специальный стандартизированный JSON объект. Для работы с API шины необходимо использовать сообщения, завернутые в подобные конверты. Большая часть информации является необязательной и потому будут описаны только необходимые поля.

Название поля	Описание
messageId	Уникальный идентификатор сообщения (необходимо генерировать при каждой отправке)
messageType	Тип сообщения. Именно этот параметр отвечает за то, как будет обрабатываться это сообщение.
message	Тело сообщения, иными словами - данные.
sentTime	Дата отправки сообщения. Данный параметр по сути ни на что не влияет, а нужен для определения возраста сообщения.

#### 5.1.2. Сообщение

Сообщение состоит из двух частей: **тип** и **тело**.

- Тип указывается в `messageType` поле конверта, он же является названием сообщения,
- Тело сообщения - это непосредственно данные сообщения для выполнения действий или с конкретной информацией о сущности, которые записываются в **message**.

### 5.2. Примеры сообщений в конверте

#### 5.2.1. Исходящее сообщение

Данный вариант конверта будет использоваться для отправки команд потребителю API Шины.

Пример конверта для исходящего сообщения [sendDocumentCommandEvent](#).

```
{
  "messageId": "ed290000-5d01-0015-6490-08d9d6d7cd93",
```

```

"messageType": "sendDocumentCommandEvent",
"message": {
  "docflowId": "22877C86-73A4-4D97-BBCF-C63A44153C8D",
  "docflowType": "ETRN",
  "senderId": "2AE4E671C65-3DAB-4974-9FF9-3AEAF80AD723",
  "receiversId": [{
    "id": "2AECD6B395E-4FCB-41AA-B10B-4E634B773772",
    "receiverType": "Carrier"
  },
  {
    "id": "2AE3DF9A700-7E0D-4B9A-A531-E1C29E1F5BC0",
    "receiverType": "Consignee"
  }
],
  "documentId": "c0478105-cd56-49e9-bb27-5786d228697d",
  "documentStorageId": "ed290100-5d01-0015-6490-08d9d6d7cd95",
  "documentName": "ON_TRNACLGROT_2AECD6B395E-4FCB-41AA-B10B-4E634B773772_2AE3DF9A700-7E0D-4B9A-A531-E1C29E1F5BC0_2AE4E671C65-3DAB-4974-9FF9-3AEAF80AD723_0_20220526_c0478105-cd56-49e9-bb27-5786d228697d.xml",
  "documentType": "ShipperInfo",
  "signatures": [{
    "id": "2AE4E671C65-3DAB-4974-9FF9-3AEAF80AD723",
    "storageId": "ed290200-5d01-0015-6490-08d9d6d7cd95",
    "fileName": "ON_TRNACLGROT_2AECD6B395E-4FCB-41AA-B10B-4E634B773772_2AE3DF9A700-7E0D-4B9A-A531-E1C29E1F5BC0_2AE4E671C65-3DAB-4974-9FF9-3AEAF80AD723_0_20220526_c0478105-cd56-49e9-bb27-5786d228697d.p7s"
  }],
  "primaryDocumentId": "c0478105-cd56-49e9-bb27-5786d228697d"
},
"sentTime": "2022-05-21T18:25:43+03:00"
}

```

### 5.2.2. Входящее сообщение

Данный вариант конверта, как правило, будет приходить в ответных сообщениях. Пример конверта для входящего сообщения [documentCameOutboundEvent](#).

```

{
  "messageId": "ed290000-5d01-0015-6490-08d9d6d7cd94",
  "messageType": "documentCameOutboundEvent",
  "message": {
    "docflowId": "22877C86-73A4-4D97-BBCF-C63A44153C8D",
    "docflowType": "ETRN",
    "senderId": "2AE4E671C65-3DAB-4974-9FF9-3AEAF80AD723",
    "receiverId": "2AE3DF9A700-7E0D-4B9A-A531-E1C29E1F5BC0",
    "documentId": "c0478105-cd56-49e9-bb27-5786d228697d",
    "documentStorageId": "ed290100-5d01-0015-6490-08d9d6d7cd95",

```

```

    "documentName": "ON_TRNACLGROT_2AEC6B395E-4FCB-41AA-B10B-4E634B773772_2AE3DF9A700-7E0D-4B9A-A531-E1C29E1F5BC0_2AE4E671C65-3DAB-4974-9FF9-3AEAF80AD723_0_20220520_c0478105-cd56-49e9-bb27-5786d228697d.xml",
    "documentType": "ShipperInfo",
    "signatures": [{
      "id": "2AE4E671C65-3DAB-4974-9FF9-3AEAF80AD723",
      "storageID": "ed290200-5d01-0015-6490-08d9d6d7cd95",
      "fileName": "ON_TRNACLGROT_2AEC6B395E-4FCB-41AA-B10B-4E634B773772_2AE3DF9A700-7E0D-4B9A-A531-E1C29E1F5BC0_2AE4E671C65-3DAB-4974-9FF9-3AEAF80AD723_0_20220520_c0478105-cd56-49e9-bb27-5786d228697d.p7s"
    }],
    "primaryDocumentId": "c0478105-cd56-49e9-bb27-5786d228697d"
  },
  "sentTime": "2022-05-21T18:35:43+03:00"
}

```

### 5.2.3. Входящее сообщение об ошибке

Данный вариант конверта, как правило, будет приходить в ответных сообщениях, в случае если конверт с ошибкой, например:

1. Неизвестный тип сообщения указанный в **messageType**

```

{
  "messageId": "ed290000-5d01-0015-6490-08d9d6d7cd94",
  "messageType": "error",
  "message": {"Неизвестный тип сообщения в messageType"},
  "sentTime": "2022-05-21T18:35:43+03:00"
}

```

2. Структура указанная в **message** не соответствует типу сообщения

```

{
  "messageId": "ed290000-5d01-0015-6490-08d9d6d7cd94",
  "messageType": "error",
  "message": {"Структура указанная в message не соответствует типу сообщения"},
  "sentTime": "2022-05-21T18:35:43+03:00"
}

```

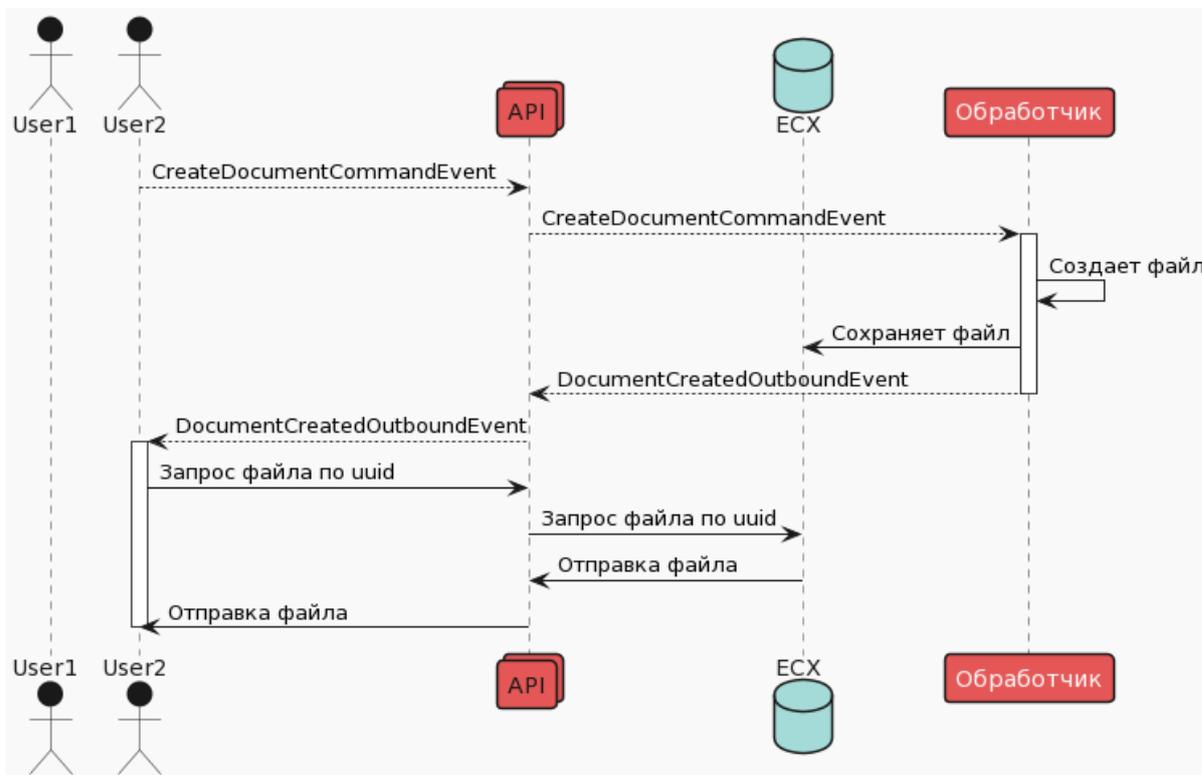
**Примечание:** Сообщения об ошибке, будет отправляться на тот же messageId, который был указан в исходящем конверте.

## 6. Сценарии использования

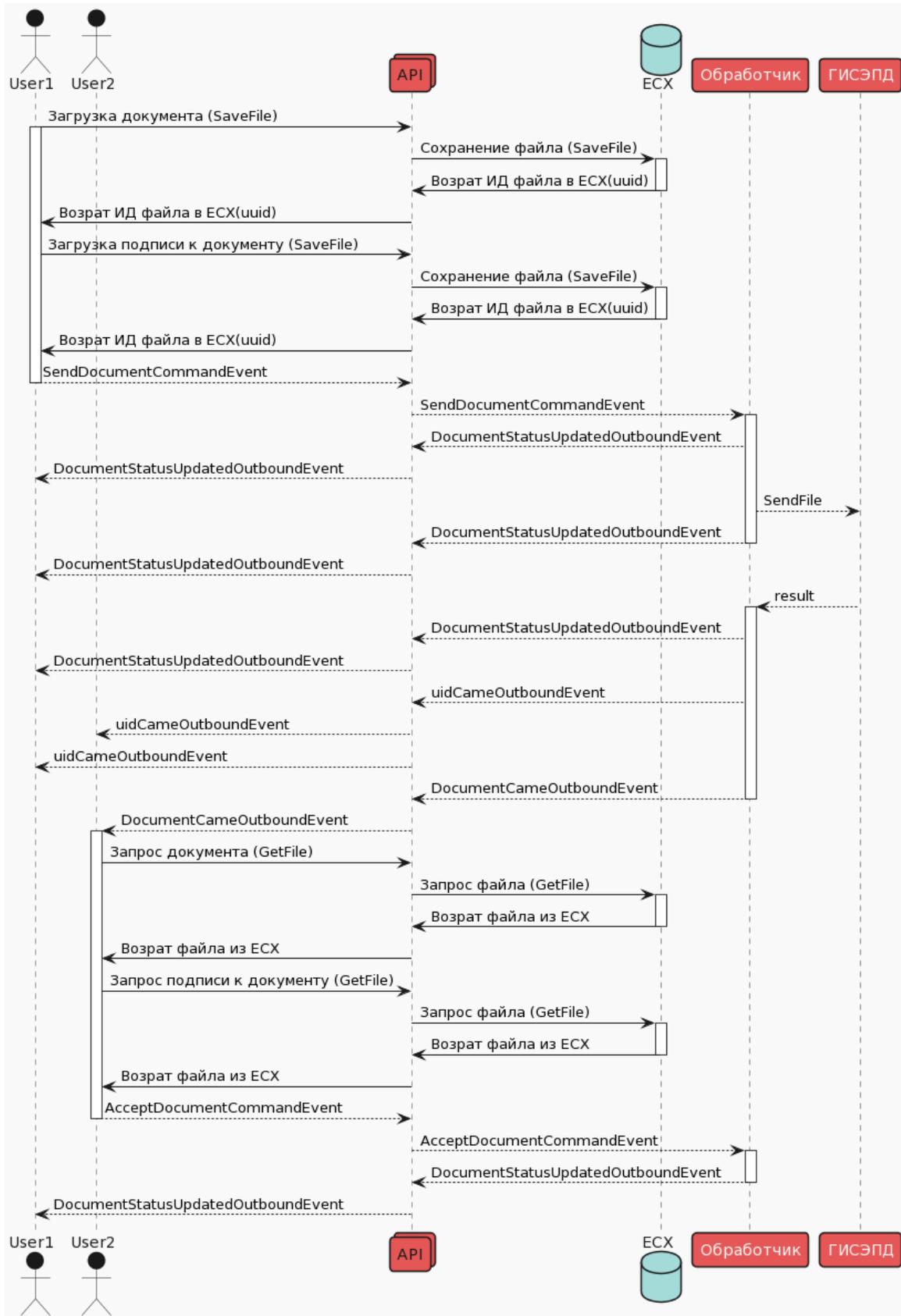
### 6.1. Документы

Данный раздел включает в себя сценарии работы с документами.

#### 6.1.1. Процесс создания служебного документа

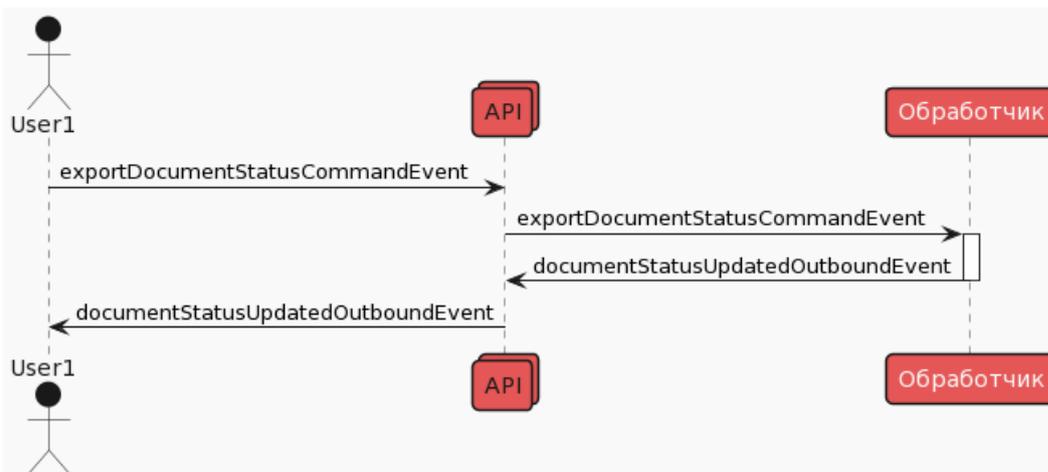


### 6.1.2. Процесс обработки документа "Информация от грузоотправителя (Т1 ЭТрН)" без УИД\_ТрН





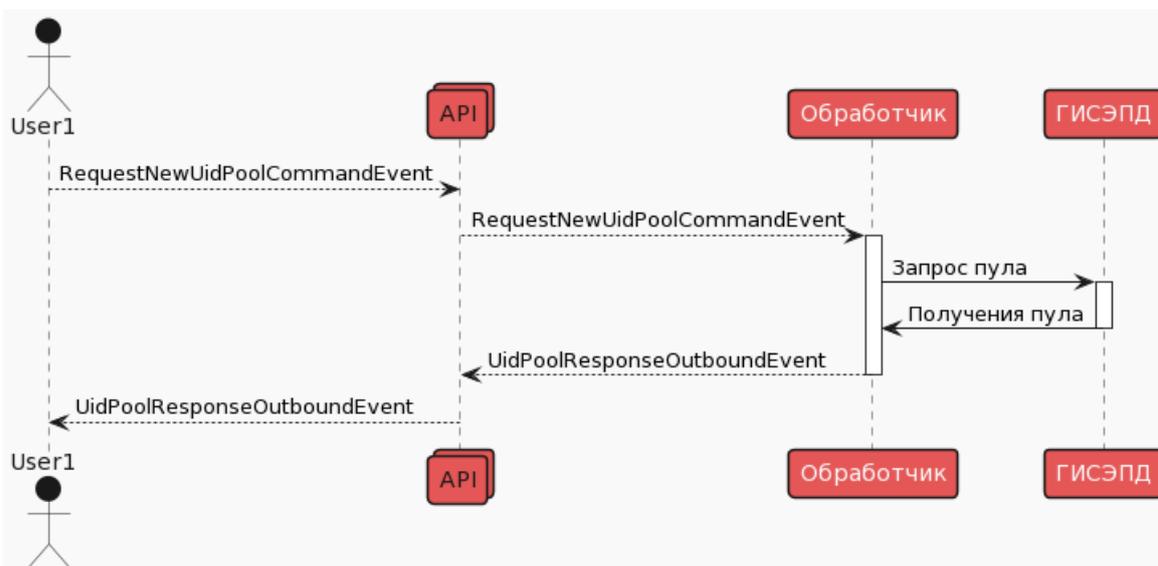
### 6.1.4. Процесс получения статуса документа



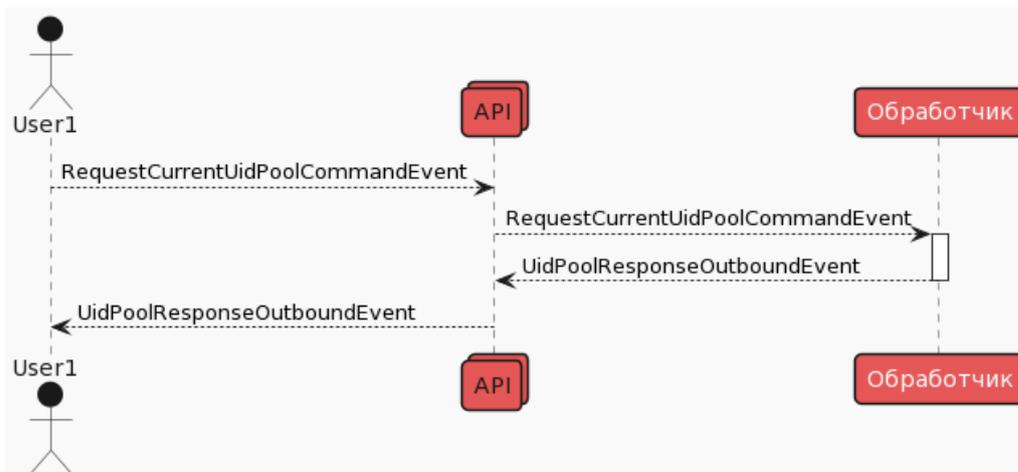
## 6.2. Уиды ГИС ЭПД

Данный раздел включает в себя сценарии работы с уидами ГИС ЭПД.

### 6.2.1. Процесс запроса пула новых уидов



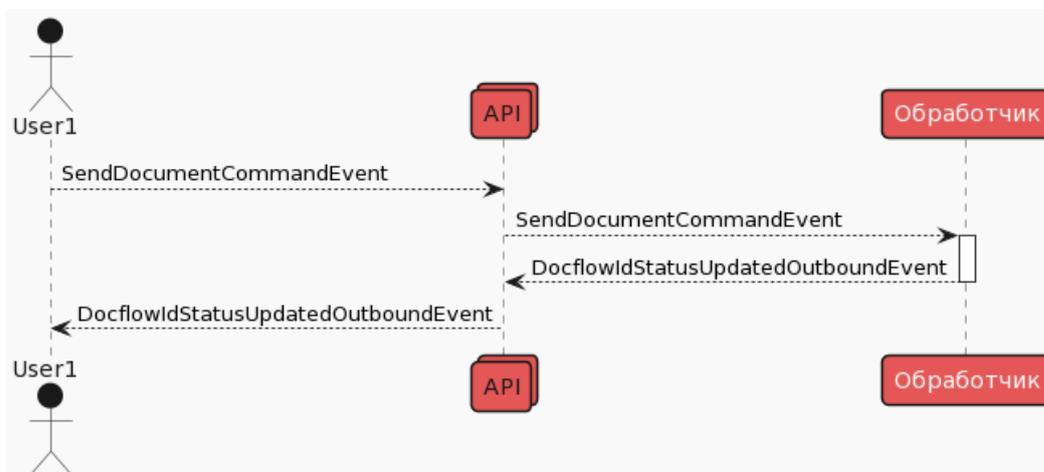
### 6.2.2. Процесс запроса пула uidов, которые ранее были выданы



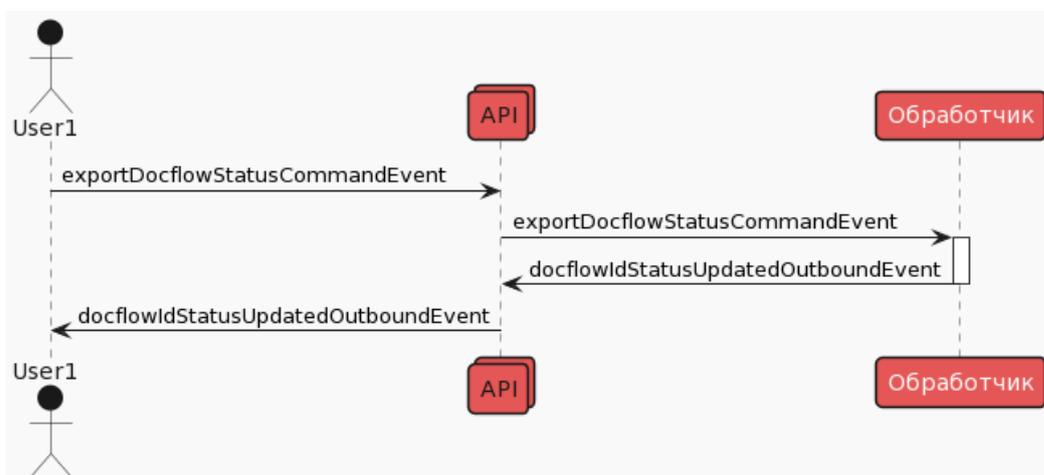
### 6.3. Документооборот

Данный раздел включает в себя сценарии работы с ДО.

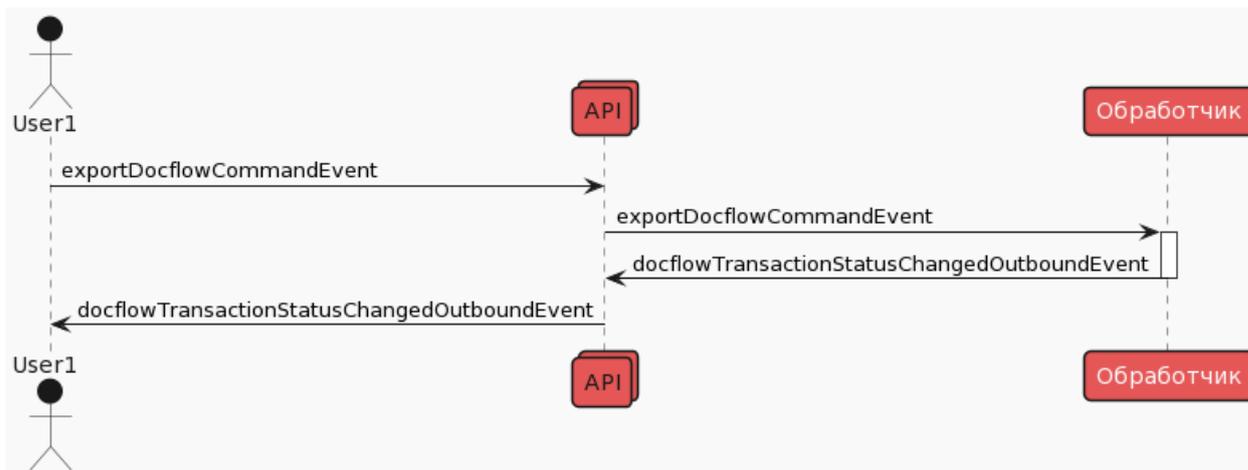
#### 6.3.1. Процесс обработки изменения состояния ДО



#### 6.3.2. Процесс получения статуса ДО



### 6.3.3. Процесс получения списка всех документов ДО

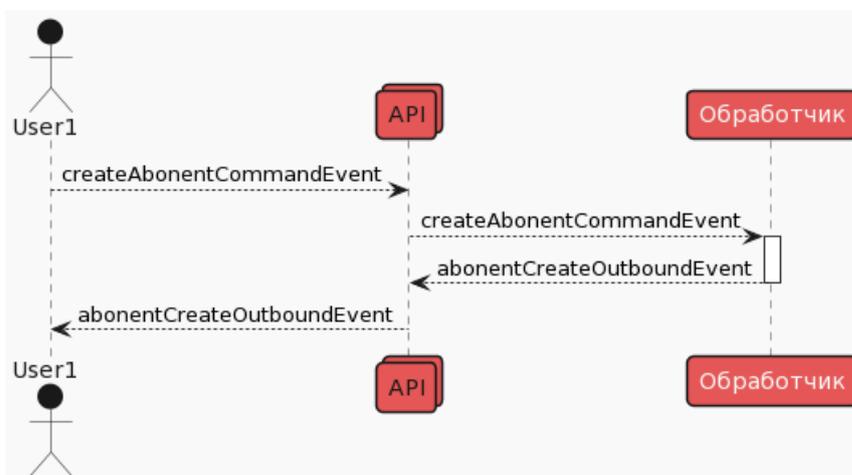


## 6.4. registration

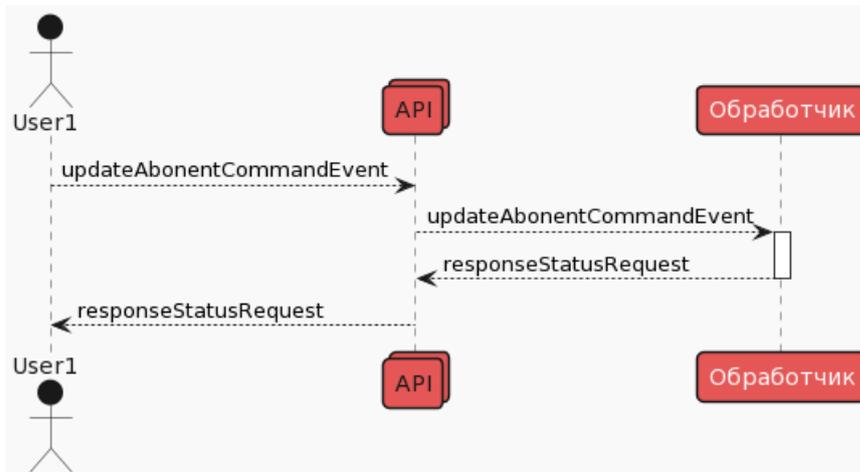
### 6.4.1. Регистрация

Данный раздел включает в себя сценарии работы с регистрацией абонентов.

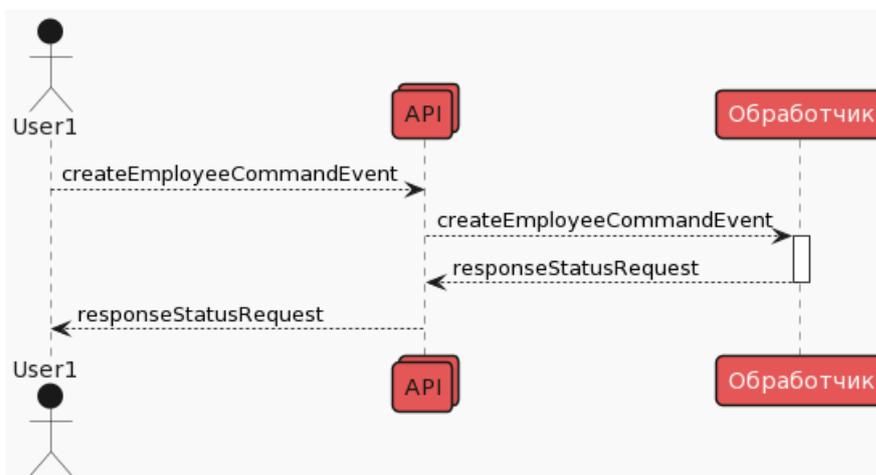
### 6.4.2. Создание абонента



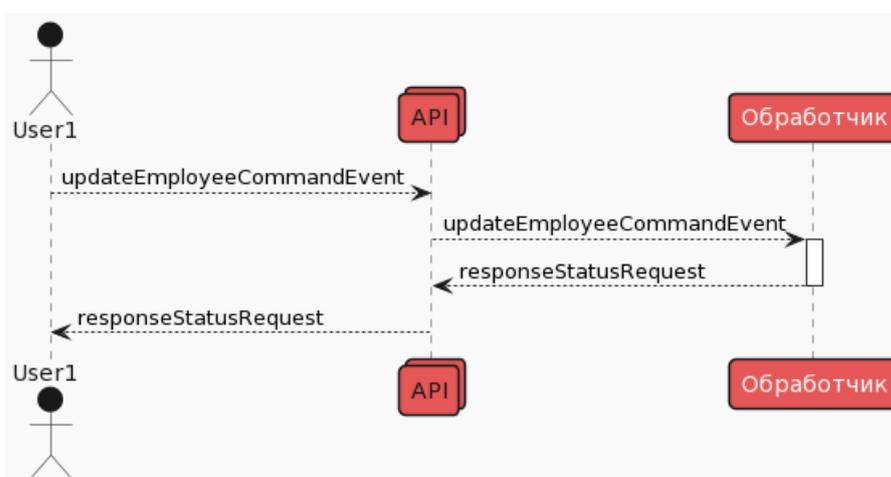
### 6.4.3. Обновление абонента



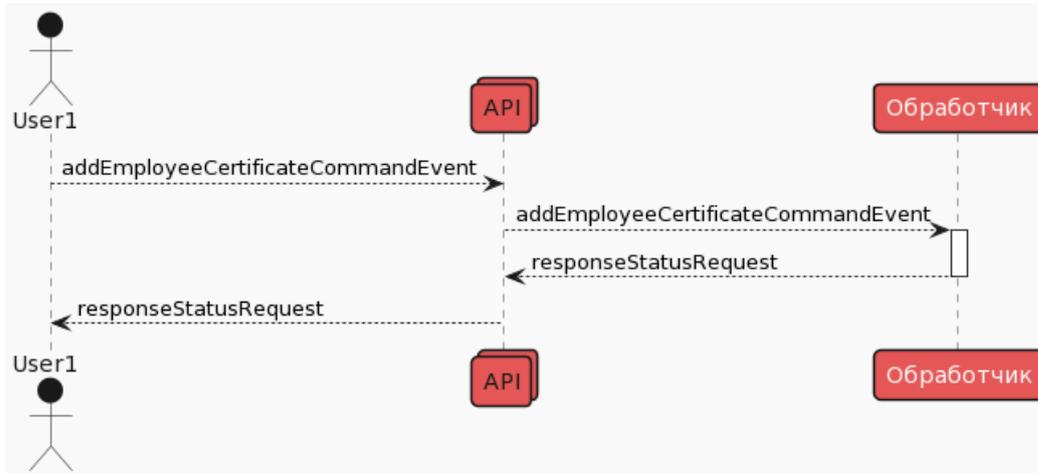
### 6.4.4. Создание сотрудника абонента



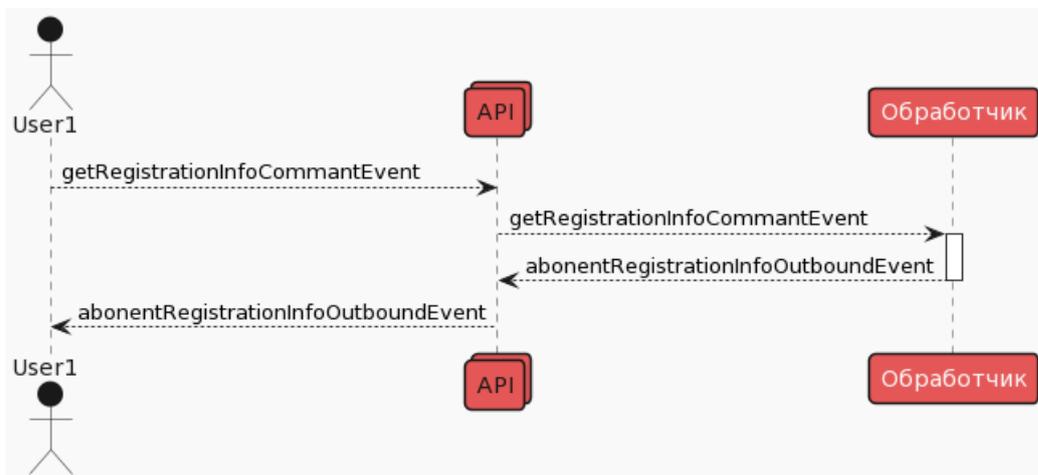
### 6.4.5. Обновление сотрудника абонента



### 6.4.6. Добавление сертификата сотрудника организации



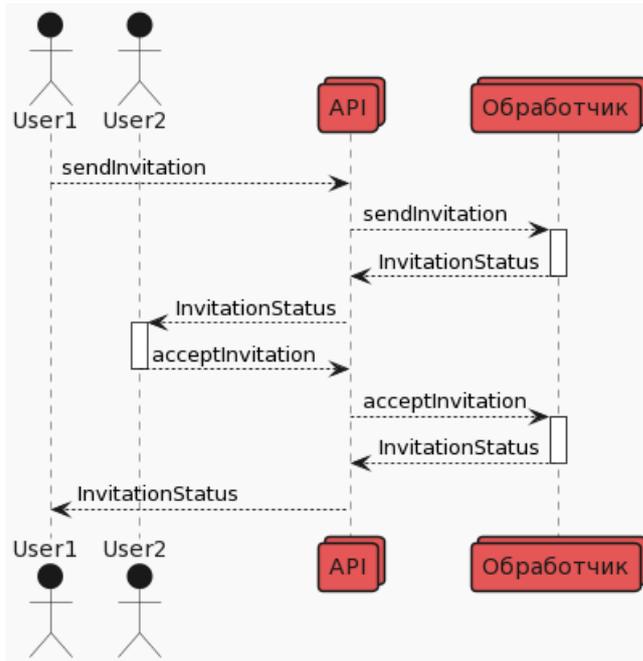
### 6.4.7. Получение данных абонента



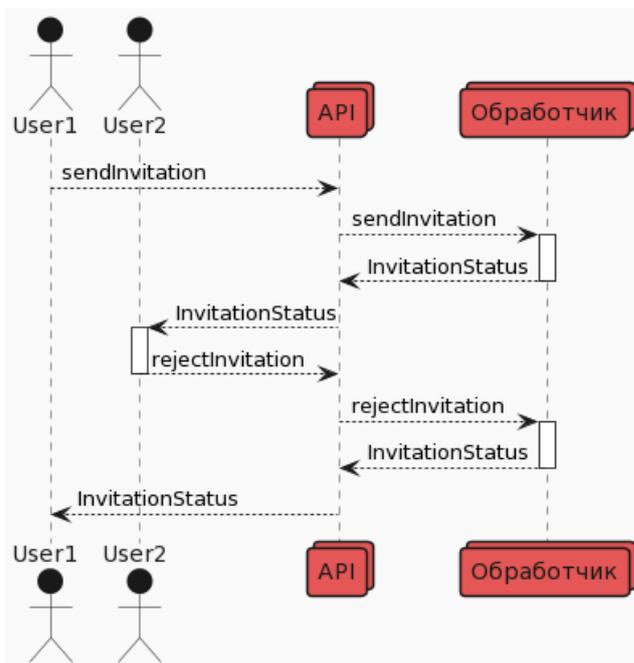
## 6.5. Приглашения

Данный раздел включает в себя сценарии работы с приглашениями.

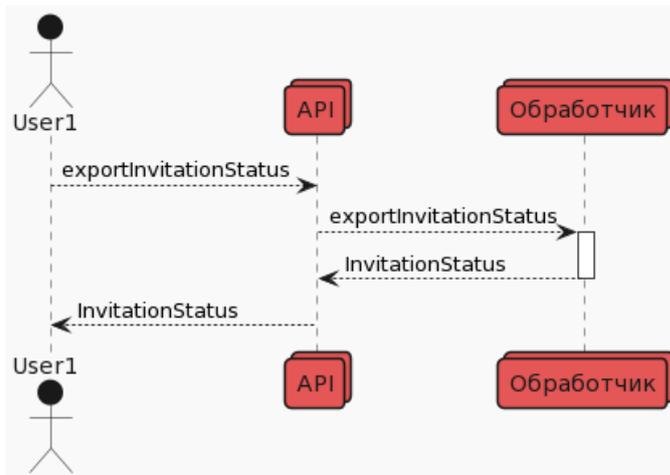
### 6.5.1. Отправить приглашение и принять приглашение



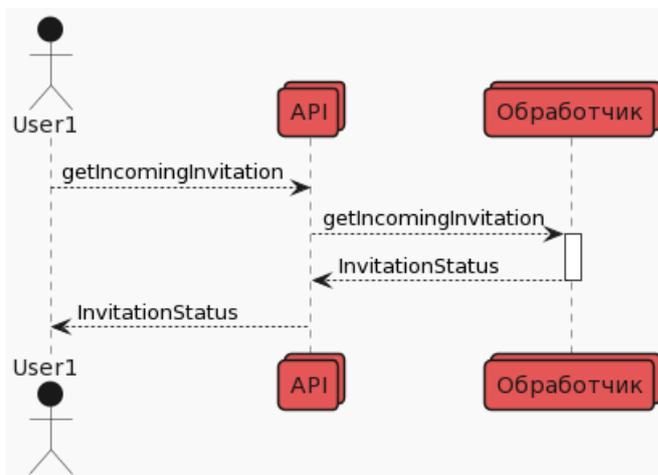
### 6.5.2. Отправить приглашение и отклонить приглашение



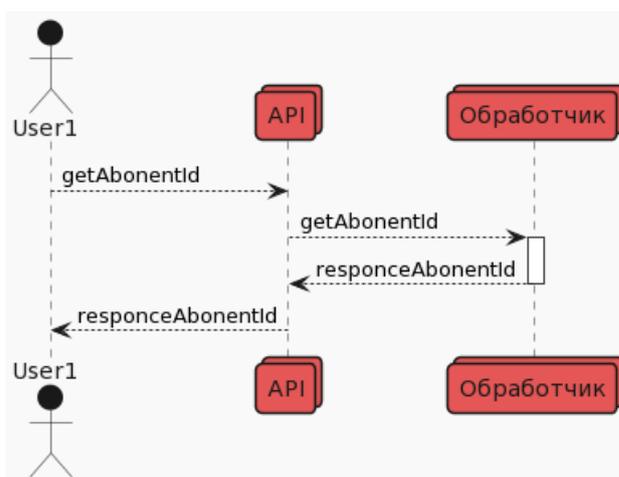
### 6.5.3. Получить статус приглашений



### 6.5.4. Получить входящие приглашения



### 6.5.5. Получить ИД ЭДО абонента



## 7. Модели сообщений

### 7.1. Документы

Группа сообщений для работы с документами.

#### 7.1.1. Командные модели

Данный блок содержит описание основных командных моделей, отвечающих за выполнение действий:

#### 4. sendDocumentCommandEvent

Команда для отправки документа.

**[Обязательно]** Документ и подпись должны быть предварительно загружены в хранилище через внешний API.

Название	Обязат.	Тип	Описание
requestId	Нет	uuid	ИД запроса (в нижнем регистре)
docflowId	Да	uuid	ИД документооборота (в нижнем регистре)
docflowType	Да	DocflowType	Тип документооборота
senderId	Да	string	ИД отправителя
senderType	Да	SenderType	Тип отправителя
receiversId	Да	ReceiverInfo[array]	ИД получателей
documentId	Да	uuid	ИД документа (в нижнем регистре)
documentStorageId	Да	uuid	ИД документа в ЕСХ оператора ЭПД (в нижнем регистре)
previousDocumentId	Нет	uuid[array]	ИД документа, в ответ на который отправляется документ / повторно отправляется документ (в нижнем регистре)
comment	Нет	string	Комментарий/причина отказа (для УОУ)

## ▼ Пример модели #1

```
{
  "docflowId": "887eca11-d711-44a0-9d00-23b25e334f9c",
  "docflowType": "ETRN",
  "senderId": "2AE4E671C65-3DAB-4974-9FF9-3AEAF80AD723",
  "receiversId": [{
    "id": "2AEC6B395E-4FCB-41AA-B10B-4E634B773772",
    "receiverType": "Carrier"
  },
  {
    "id": "2AE3DF9A700-7E0D-4B9A-A531-E1C29E1F5BC0",
    "receiverType": "Consignee"
  }
  ],
  "documentId": "c0478105-cd56-49e9-bb27-5786d228697d",
  "documentStorageId": "ed290100-5d01-0015-6490-08d9d6d7cd95",
  "documentName": "ON_TRNACLGROT_2AEC6B395E-4FCB-41AA-B10B-4E634B773772_2AE3DF9A700-7E0D-4B9A-A531-E1C29E1",
  "documentType": "ShipperInfo",
  "signatures": [{
    "id": "2AE4E671C65-3DAB-4974-9FF9-3AEAF80AD723",
    "storageID": "ed290200-5d01-0015-6490-08d9d6d7cd95",
    "fileName": "ON_TRNACLGROT_2AEC6B395E-4FCB-41AA-B10B-4E634B773772_2AE3DF9A700-7E0D-4B9A-A531-E1C29E1"
  }]
}
```

## ▼ Пример модели #2

```
{
  "docflowId": "887eca11-d711-44a0-9d00-23b25e334f9c",
  "docflowType": "ETRN",
  "senderId": "2AEC6B395E-4FCB-41AA-B10B-4E634B773772",
  "receiversId": [{
    "id": "2AE4E671C65-3DAB-4974-9FF9-3AEAF80AD723",
    "receiverType": "Consignor"
  },
  {
    "id": "2AE3DF9A700-7E0D-4B9A-A531-E1C29E1F5BC0",
    "receiverType": "Consignee"
  }
  ],
  "documentId": "829e4148-15a9-36c5-c971-bc4b2dc10434",
  "documentStorageId": "ed290100-5d01-0015-6490-08d9d6d7cd95",
  "documentName": "ON_TRNACLPPRIN_2AE3DF9A700-7E0D-4B9A-A531-E1C29E1F5BC0_2AE4E671C65-3DAB-4974-9FF9-3AEAF8",
  "documentType": "CarrierInfoAcceptanceOfCargo",
  "signatures": [{
    "id": "2AE4E671C65-3DAB-4974-9FF9-3AEAF80AD723",
    "storageID": "ed290200-5d01-0015-6490-08d9d6d7cd95",
    "fileName": "ON_TRNACLPPRIN_2AE3DF9A700-7E0D-4B9A-A531-E1C29E1F5BC0_2AE4E671C65-3DAB-4974-9FF9-3AEAF8"
  }],
  "previousDocumentId": [
    "c0478105-cd56-49e9-bb27-5786d228697d"
  ]
}
```

## 5. createDocumentCommandEvent

## Команда для создания документа.

Название	Обязат.	Тип	Описание
requestId	Да	uuid	ИД запроса (в нижнем регистре)
documentType	Да	DocTypeService	Тип сервисного документа
senderId	Да	string	ИД отправителя
receiverId	Да	string	ИД получателя
signatory	Да	SignatoryInfo	Данные подписанта
primaryDocumentId	Нет	uuid	ИД документа, на который формируется ИОП/УОУ (в нижнем регистре)
comment	Нет	string	Комментарий/причина отказа (для УОУ). Для УОУ обязательное поле.

## ▼ Пример модели

```
{
  "requestId": "ebd0f163-4586-4e7e-a23c-70f9c5f0886e0",
  "documentType": "ИОП",
  "senderId": "2AEF9596552-364A-402E-AFFE-7C01F9C1B449",
  "receiverId": "2AE09596552-364A-402E-AFFE-7C01F9C1B449",
  "signatory": {
    "name": "Иван",
    "surname": "Иванов",
    "patronymic": "Иванович",
    "post": "Директор"
  },
  "primaryDocumentId": "98c0be43-ca0e-48fa-b1ef-fe6d55efbfec"
}
```

## 6. acceptDocumentCommandEvent

## Команда принятия документа (документ получен абонентом).

Название	Обязат.	Тип	Описание
requestId	Нет	uuid	ИД запроса (в нижнем регистре)
docflowId	Да	uuid	ИД документооборота (в нижнем регистре)
documentId	Да	uuid	ИД документа (в нижнем регистре)
receiversId	Да	ReceiverInfo[array]	ИД получателя
errorMessage	Нет	string	Описание ошибки, почему не принят документ. Заполняется только если не удалось принять документ

## ▼ Пример модели

```
{
  "docflowId": "887eca11-d711-44a0-9d00-23b25e334f9c",
  "documentId": "829e4148-15a9-36c5-c971-bc4b2dc10434",
  "receiversId": [{
    "id": "2AEC6B395E-4FCB-41AA-B10B-4E634B773772",
    "receiverType": "Carrier"
  },
  {
    "id": "2AE3DF9A700-7E0D-4B9A-A531-E1C29E1F5BC0",
    "receiverType": "Consignee"
  }
  ]
}
```

## 7. exportDocumentStatusCommandEvent

Команда для запроса текущего статуса документа.

Название	Обязат.	Тип	Описание
requestId	Да	uuid	ИД исходного запроса (в нижнем регистре)
documentId	Да	uuid	ИД документа (в нижнем регистре)

## ▼ Пример модели

```
{
  "requestId": "abf003ca-6433-4535-87b8-dcb809b3d628",
  "documentId": "92722ee6-82d3-4926-98c4-2a207846eec3"
}
```

## 7.1.2. Ответные модели

Данный блок содержит описание основных ответных моделей от ядра системы.

## 1. documentCameOutboundEvent

Событие о приходе документа в сторону подписчика шины.

**[Обязательно]** Документ и подпись должны быть предварительно загружены в хранилище через внешний API.

Название	Обязат.	Тип	Описание
requestId	Нет	uuid	ИД запроса (в нижнем регистре)
docflowId	Да	uuid	ИД документооборота (в нижнем регистре)
docflowType	Да	DocflowType	Тип докуменооборота
senderId	Да	string	ИД отправителя
senderType	Да	SenderType	Тип отправителя
receiversId	Да	ReceiverInfo[array]	Информация о получателях
documentId	Да	uuid	ИД документа (в нижнем регистре)
documentStorageId	Да	uuid	ИД документа в ЕСХ оператора ЭПД (в нижнем регистре)
documentName	Да	string	Наименование файла документа
documentType	Да	DocType	Тип документа
signatures	Нет	SignatureInfo[array]	Подписи к документу
primaryDocumentId	Нет	uuid	ИД первичного документа в ДО
previousDocumentId	Нет	uuid[array]	ИД документа, в ответ на который отправляется документ / повторно отправляется документ (в нижнем регистре)
comment	Нет	string	Комментарий/причина отказа (для УОУ)
transactionId	да	string	Наименование транзакции

## ▼ Пример модели

```

{
  "docflowId": "887eca11-d711-44a0-9d00-23b25e334f9c",
  "docflowType": "ETRN",
  "senderId": "2AE4E671C65-3DAB-4974-9FF9-3AEAF80AD723",
  "receiversId": [{
    "id": "2AEC6B395E-4FCB-41AA-B10B-4E634B773772",
    "receiverType": "Carrier"
  },
  {
    "id": "2AE3DF9A700-7E0D-4B9A-A531-E1C29E1F5BC0",
    "receiverType": "Consignee"
  }
  ],
  "documentId": "c0478105-cd56-49e9-bb27-5786d228697d",
  "documentStorageId": "ed290100-5d01-0015-6490-08d9d6d7cd95",
  "documentName": "ON_TRNACLGROT_2AEC6B395E-4FCB-41AA-B10B-4E634B773772_2AE3DF9A700-7E0D-4B9A-A531-E1C29E1F5BC",
  "documentType": "ShipperInfo",
  "signatures": [{
    "id": "2AE4E671C65-3DAB-4974-9FF9-3AEAF80AD723",
    "storageID": "ed290200-5d01-0015-6490-08d9d6d7cd95",
    "fileName": "ON_TRNACLGROT_2AEC6B395E-4FCB-41AA-B10B-4E634B773772_2AE3DF9A700-7E0D-4B9A-A531-E1C29E1F5BC"
  }]
}

```

## 2. documentCreatedOutboundEvent

Событие оповещения создания документа.

Название	Обязат.	Тип	Описание
requestId	Да	uuid	ИД исходного запроса (в нижнем регистре)
documentStorageId	Да	uuid	ИД документа в ЕСХ оператора ЭПД (в нижнем регистре)
errorMessage	Нет	string	Описание ошибки, почему не создан документ. Заполняется только если не удалось создать документ

## ▼ Пример модели

```

{
  "requestId": "3b0fe459-b7cf-4a9c-98ae-a5850e2821da",
  "documentStorageId": "92722ee6-82d3-4926-98c4-2a207846eec3"
}

```

### 3. documentStatusUpdatedOutboundEvent

Событие оповещения изменения статуса документа.

Название	Обязат.	Тип	Описание
requestId	Да	uuid	ИД исходного запроса (в нижнем регистре)
documentId	Да	uuid	ИД документа (в нижнем регистре)
status	Да	StatusType	Текущий статус документа
nextStatus	Нет	StatusType	Ожидаемый статус документа
errorMessage	Нет	string	Содержит сообщение об ошибке (заполняется только если status = Error)

#### ▼ Пример модели

```
{
  "requestId": "92eec270-0590-4f5c-b0dc-4dc8ff360d35",
  "documentId": "92722ee6-82d3-4926-98c4-2a207846eec3",
  "status": "Initialized",
  "nextStatus": "Checking"
}
```

### 7.1.3. Дополнительные модели

Данный блок содержит описание дополнительных моделей.

#### 1. DocflowType

Тип ДО

Значение	Описание
ETRN	Электронная транспортная накладная
ESV	Электронная сопроводительная ведомость
EZN	Электронный заказ-наряд
Unknown	Тип неизвестен

## 2. StatusType

Статусы документов.

Значение	Описание
Initialized	Документ получен оператором
Checking	Выполняется проверка
Verified	Документ принят оператором
SentToGis	Документ отправлен в ГИС
AcceptedToGis	Документ принят в ГИС
SentToRecipient	Документ отправлен получателю
Delivered	Документ доставлен получателю
Error	Ошибка по документу
Unknown	Тип неизвестен.

## 3. ReceiverInfo

Информация о получателе.

Название	Обязат.	Тип	Описание
id	Да	string	Идентификатора абонента внутри системы.
receiverType	Да	ReceiverType	Тип получателя

#### 4. ReceiverType

Типы получателей.

Значение	Описание
Consignor	Грузоотправитель
Carrier	Перевозчик
Consignee	Грузополучатель
Charterer	Фрахтователь
Freighter	Фрахтовщик
AnotherReceiver	Иной получатель
Unknown	Тип неизвестен.

#### 5. SenderType

Типы отправителей.

Значение	Описание
Consignor	Грузоотправитель
Carrier	Перевозчик
Consignee	Грузополучатель
Charterer	Фрахтователь
Freighter	Фрахтовщик
Unknown	Тип неизвестен.

## 6. SignatureInfo

Информация по подписи.

Название	Обязат.	Тип	Описание
id	Да	string(uuid)	ИД подписи
storageID	Да	string(uuid)	ИД в ЕСХ оператора ЭПД
fileName	Да	string	Имя файла подписи

## 7. SignatoryInfo

Информация по подписанту.

Название	Обязат.	Тип	Описание
name	Да	string	Имя подписанта
surname	Да	string	Фамилия подписанта
patronymic	Нет	string	Отчество подписанта
post	Да	string	Должность подписанта

## 8. DocTypeService

Типы сервисных документов.

Значение	Описание
IOP	Извещение об получении
UOU	Уведомление об уточнении

## 9. DocType

## Типы документов.

Значение	Описание
ShipperInfo	Информация от грузоотправителя (Т1 ЭТрН)
CarrierInfoAcceptanceOfCargo	Информация от перевозчика о получении груза (Т2 ЭТрН)
ConsigneeInfo	Информация от грузополучателя (Т3 ЭТрН)
CarrierInfoDeliveryOfCargo	Информация от перевозчика о выдаче груза (Т4 ЭТрН)
ReaddressingInfo	Информация о переадресовке (Т7 ЭТрН)
ReplacementInfo	Информация о замене водителя (водителей) и (или) транспортного средства (Т8 ЭТрН)
CarrierFinancialConditionChange	Информация об изменении финансового состояния перевозчика (Т5 ЭТрН)
ShipperFinancialConditionChangeConfirmation	Подтверждение грузоотправителя об изменении финансового состояния (Т6 ЭТрН)
ChartererInformation	Файл обмена информации фрахователя (Т1 ЭЭН)
FreighterInformation	Файл обмена информации фраховщика (Т2 ЭЭН)
ChartererInformationDeliveryTransport	Файл обмена информации фрахователя по факту подачи транспортного средства (Т3 ЭЭН)
FreighterInformationReturnTransport	Файл обмена информации фраховщика по факту возврата транспортного средства (Т4 ЭЭН)
CarrierInformation_1	Файл обмена информации перевозчика при передаче порожнего контейнера грузоотправителю Т1-1 ON_SOPVEDPER1
ShipperInformation_1	Файл обмена информации грузоотправителя при получении порожнего контейнера от перевозчика Т2-1 ON_SOPVEDGO1
ShipperInformation_2	Файл обмена информации грузоотправителя при передаче груженого контейнера перевозчику Т2-2 ON_SOPVEDGO2
CarrierInformation_2	Файл обмена информации перевозчика при получении груженого контейнера от грузоотправителя Т1-2 ON_SOPVEDPER2
CarrierInformation_3	Файл обмена информации перевозчика при передаче груженого контейнера грузополучателю Т1-3 ON_SOPVEDPER3
ConsigneeInformation_1	Файл обмена информации грузополучателя при получении груженого контейнера от перевозчика Т3-1 ON_SOPVEDGP1
ConsigneeInformation_2	Файл обмена информации грузополучателя при подаче транспортного средства перевозчиком для погрузки порожнего контейнера Т3-2 ON_SOPVEDGP2

CarrierInformation_4	Файл обмена информации перевозчика при получении порожнего контейнера от грузополучателя T1-4 ON_SOPVEDPER4
CarrierInformation_5	Файл обмена информации перевозчика при подаче транспортного средства для погрузки порожнего контейнера T1-5 ON_SOPVEDPER5
IOP	Извещение об получении
UOU	Уведомление об уточнении

## 7.2. Документооборот

Группа сообщений для работы с документооборотом.

### 7.2.1. Командные модели

Данный блок содержит описание основных командных моделей, отвечающих за выполнение действий.

#### 1. exportDocflowStatusCommandEvent

Команда получения текущего статуса документооборота.

Название	Обязат.	Тип	Описание
requestId	Да	uuid	ИД исходного запроса (в нижнем регистре)
abonentId	Да	string	ИД абонента
docflowId	Да	uuid	ИД документооборота (в нижнем регистре)

#### ▼ Пример модели

```
{
  "requestId": "a8b53313-4603-4988-a91b-b313d262a6a6",
  "abonentId": "2AE09596552-364A-402E-AFFE-7C01F9C1B449",
  "docflowId": "a8b53313-4603-4988-a91b-b313d262a6a9"
}
```

## 2. exportDocflowCommandEvent

Команда получения всех документов в указанном документообороте.

Название	Обязат.	Тип	Описание
requestId	Да	uuid	ИД исходного запроса (в нижнем регистре)
abonentId	Да	string	ИД абонента
docflowId	Да	uuid	ИД документооборота (в нижнем регистре)

### ▼ Пример модели

```
{
  "requestId": "a8b53313-4603-4988-a91b-b313d262a6a6",
  "abonentId": "2AE09596552-364A-402E-AFFE-7C01F9C1B449",
  "docflowId": "a8b53313-4603-4988-a91b-b313d262a6a9"
}
```

## 7.2.2. Ответные модели

Данный блок содержит описание основных ответных моделей от ядра системы.

### 1. docflowIdStatusUpdatedOutboundEvent

Событие оповещения о текущем статусе.

Название	Обязат.	Тип	Описание
requestId	Да	uuid	ИД исходного запроса (в нижнем регистре)
abonentId	Да	string	ИД абонента
docflowId	Да	uuid	ИД документооборота (в нижнем регистре)
status	Да	StatusTypeDocflow	Текущий статус документооборота
allowedActions	Нет	DocflowAllowedActions[array]	Доступные действия с документооборотом
errorMessage	Нет	string	Содержит сообщение об ошибке (заполняется только если status = Error)

## ▼ Пример модели

```
{
  "requestId": "a8b53313-4603-4988-a91b-b313d262a6a6",
  "abonentId": "2AE09596552-364A-402E-AFFE-7C01F9C1B449",
  "docflowId": "92722ee6-82d3-4926-98c4-2a207846eec3",
  "status": "Pending",
  "allowedActions": [
    "AcceptCargoDelivery",
    "AcceptCargoDeliveryIssue",
    "IOP",
    "UOU"
  ]
}
```

## 2. docflowTransactionStatusChangedOutboundEvent

Событие оповещения о всех документах в указанном документообороте.

Название	Обязат.	Тип	Описание
requestId	Да	uuid	ИД исходного запроса (в нижнем регистре)
abonentId	Да	string	ИД абонента
docflowId	Да	uuid	ИД документооборота (в нижнем регистре)
documents	Да	DocumentInfo[array]	Информация о документах указанного документооборота

## ▼ Пример модели

```
{
  "requestId": "a8b53313-4603-4988-a91b-b313d262a6a6",
  "abonentId": "2AE09596552-364A-402E-AFFE-7C01F9C1B449",
  "docflowId": "a8b53313-4603-4988-a91b-b313d262a6a0",
  "documents": [
    {
      "id": "a8b53313-4603-4988-a91b-b313d262a6a6",
      "documentName": "ON_TRNACLGROT_2AE6468B9F2-A7D4-4B17-AA81-89C1DD7D29ED_2AEAAA9E60D-9C91-4E6E-889B",
      "documentStorageId": "a8b53313-4603-4988-a91b-b313d262a6a6",
      "documentType": "ShipperInfo"
    },
    {
      "id": "a8b53313-4603-4988-a91b-b313d262a6a6",
      "documentName": "ON_TRNACLPPRIN_2AEAAA9E60D-9C91-4E6E-889B-C2365110B24C_2AE7423D076-051C-42BE-B3E",
      "documentStorageId": "a8b53313-4603-4988-a91b-b313d262a6a6",
      "documentType": "CarrierInfoAcceptanceOfCargo"
    }
  ]
}
```

### 7.2.3. Дополнительные модели

Данный блок содержит описание дополнительных моделей.

#### 1. StatusTypeDocflow

Статусы документооборота.

Значение	Описание
Pending	В обработке
AwaitingShipment	Ожидает отгрузки со склада отправителя
AcceptCargoDelivery	Груз принят к перевозке
AcceptCargoDeliveryIssue	Груз принят к перевозке с замечаниями
AcceptCargoConsignee	Груз принят грузополучателем
AcceptCargoConsigneeIssue	Груз принят грузополучателем с замечаниями
RejectCargoConsignee	Отказано в приемке груза
ReleasedCargoCarrier	Груз выдан грузополучателю
ReleasedCargoCarrierIssue	Груз выдан грузополучателю с замечаниями
FinancialChangeDelivery	Расчет стоимости перевозки
ChangeDeliveryPoint	Переадресовка
ReplacementInfo	Замена водителя / транспортного средства
Finished	Документооборот завершен
VehicleRequest	Запрос транспортного средства
VehicleRequestConfirmed	Запрос транспортного средства подтвержден
DeliveryVehicle	Подача транспортного средства
VehicleReturn	Возврат транспортного средства
TransferShipperEmptyContainer	Передача порожнего контейнера грузоотправителю
ReceiptFromCarrierEmptyContainer	Получение порожнего контейнера от перевозчика

DeliveryVehicleEmptyContainer	Подача транспортного средства под погрузку порожнего контейнера
TransferToCarrierEmptyContainer	Передача порожнего контейнера перевозчику
ReceiptFromConsigneeEmptyContainer	Получение порожнего контейнера от грузополучателя
ReceiptFromShipperLoadedContainer	Получение груженого контейнера от грузоотправителя
TransferToCarrierLoadedContainer	Передача груженого контейнера перевозчику
TransferToConsigneeLoadedContainer	Передача груженого контейнера грузополучателю
ReceiptFromCarrierLoadedContainer	Получение груженого контейнера от перевозчика
UOU	Уведомление об уточнении
CorrectionRequired	Требуется исправление документа(ГИС)
Error	Ошибка по документу
Unknown	Тип неизвестен

## 2. DocflowAllowedActions

Список доступных действий для данного документооборота.

Значение	Описание
AcceptCargoDelivery	Принять груз к перевозке
AcceptCargoDeliveryIssue	Принять груз к перевозке с замечаниями
AcceptCargoConsignee	Принять груз грузополучателем
AcceptCargoConsigneeIssue	Принять груз грузополучателем с замечаниями
RejectCargoConsignee	Отказать в приемке груза
ReleasedCargoCarrier	Выдать груз грузополучателю
ReleasedCargoCarrierIssue	Выдать груз грузополучателю с замечаниями
SendFinancialChangeDelivery	Отправить изменение стоимости перевозки
FinancialChangeDelivery	Принять изменение стоимость перевозки

ChangeDeliveryPoint	Оформить переадресовку
ReplacementInfo	Оформить изменения в перевозке
VehicleRequest	Подтвердить запрос на транспортное средство
VehicleRequestConfirmed	Оформить подачу транспортного средства
VehicleReturn	Оформить возврат транспортного средства
TransferShipperEmptyContainer	Передать порожний контейнер грузоотправителю
ReceiptFromCarrierEmptyContainer	Принять порожний контейнер от перевозчика
DeliveryVehicleEmptyContainer	Подать транспортное средство под погрузку порожнего контейнера
TransferToCarrierEmptyContainer	Передать порожний контейнер перевозчику
ReceiptFromConsigneeEmptyContainer	Принять порожний контейнер от грузополучателя
ReceiptFromShipperLoadedContainer	Принять груженный контейнер от грузоотправителя
TransferToCarrierLoadedContainer	Передать груженный контейнер перевозчику

SendIOP	Отправить уведомление о получении
SendUOU	Отправить уведомление об уточнении
Correction	Требуется исправление документа(ГИС)
Unknown	Тип неизвестен

### 3. DocumentInfo

Информация по документу.

Название	Обязат.	Тип	Описание
id	<b>Да</b>	uuid	ИД документа (в нижнем регистре)
documentName	<b>Да</b>	string	Наименование файла
documentStorageId	<b>Да</b>	uuid	ИД документа в ЕСХ оператора ЭПД (в нижнем регистре)
documentType	<b>Да</b>	DocType	Тип документа
signatures	<b>Нет</b>	SignatureInfo [array]	Тип документа

## 4. SignatureInfo

Информация о подписи.

Название	Обязат.	Тип	Описание
storageID	Да	uuid	ИД в ЕСХ оператора ЭПД (в нижнем регистре)
fileName	Да	string	Имя файла подписи

### 7.3. УИДы

Группа сообщений для работы с УИДами ГИС ЭПД.

#### 7.3.1. Командные модели

Данный блок содержит описание основных командных моделей, отвечающих за выполнение действий.

##### 1. requestNewUidPoolCommandEvent

Команда для получения новых зарезервированных УИДов для абонента.

Название	Обязат.	Тип	Описание
requestId	Да	uuid	ИД исходного запроса (в нижнем регистре)
abonentId	Да	string	ИД абонента
count	Нет	int	Количество УИДов, если не указывать, то значение = 1

##### ▼ Пример модели

```
{
  "requestId": "a8b53313-4603-4988-a91b-b313d262a6a6",
  "abonentId": "2AE09596552-364A-402E-AFFE-7C01F9C1B449"
}
```

## 2. requestCurrentUidPoolCommandEvent

Команда для получения полученных УИДов по абоненту.

Название	Обязат.	Тип	Описание
requestId	<b>Да</b>	uuid	ИД исходного запроса (в нижнем регистре)
abonentId	<b>Да</b>	string	ИД абонента
limit	<b>Нет</b>	int	Запрашиваемое количество УИДов (максимальное количество 1000). Если не указать, то все свободные.

### ▼ Пример модели

```
{
  "requestId": "a8b53313-4603-4988-a91b-b313d262a6a6",
  "abonentId": "2AE09596552-364A-402E-AFFE-7C01F9C1B449"
}
```

### 7.3.2. Ответные модели

Данный блок содержит описание основных ответных моделей от ядра системы.

#### 1. uidPoolResponseOutboundEvent

Событие возвращает список УИДов для абонента.

Название	Обязат.	Тип	Описание
requestId	<b>Да</b>	uuid	ИД исходного запроса (в нижнем регистре)
abonentId	<b>Да</b>	string	ИД абонента
typeUid	<b>Нет</b>	TypeUid	Тип УИДа
uids	<b>Нет</b>	uuid[array]	УИД выданные ГИСом и зарезервированные за абонентом
message	<b>Нет</b>	string	Сообщение абоненту

## ▼ Пример модели

```
{
  "requestId": "a8b53313-4603-4988-a91b-b313d262a6a6",
  "abonentId": "2AE09596552-364A-402E-AFFE-7C01F9C1B449",
  "typeUid": "NewUid",
  "uids": [
    "a8b53313-4603-4988-a91b-b313d262a6a6",
    "a8b53313-4603-4988-a91b-b313d262a6a7",
    "a8b53313-4603-4988-a91b-b313d262a6a8",
    "a8b53313-4603-4988-a91b-b313d262a6a9"
  ],
  "message": "Количество выдаваемых УИД 4 ограничено относительно запрошенных 20 с учетом выданных ранее и неи
}
```

## 2. uidCameOutboundEvent

Событие о приходе УИДа от ГИС ЭПД по конкретному ДО в сторону подписчика ШИНЫ.

Название	Обязат.	Тип	Описание
docflowId	Да	uuid	ИД документооборота (в нижнем регистре)
receiverId	Да	string	ИД получателя
documentId	Да	uuid	ИД документа
uid	Да	uuid	УИД

## ▼ Пример модели

```
{
  "docflowId": "a8b53313-4603-4988-a91b-b313d262a6a6",
  "receiverId": "2AEK9596552-364A-402E-AFFE-7C01F9C1B420",
  "documentId": "a8b53313-4603-4988-a91b-b313d262a6a7",
  "uid": "a8b53313-4603-4988-a91b-b313d262a6a6"
}
```

### 7.3.3. Дополнительные модели

Данный блок содержит описание дополнительных моделей.

#### 1. StatusType

Типы УИДов.

Значение	Описание
NewUid	УИД в ответ на команду получить новые УИДы
CurrentUid	УИД в ответ на команду получить все использованные УИДы
Unknown	Тип неизвестен

### 7.4. Ошибки

Группа сообщений с ошибками для внешнего мира.

#### 7.4.1. Ответные модели

Данный блок содержит описание основных ответных моделей от ядра системы.

#### 1. errorEvent

Событие возвращает ошибку и возможно ее описание.

Данная ошибка предназначена только для вывода транспортных ошибок из шины клиенту.

Название	Обязат.	Тип	Описание
requestId	Да	uuid	ИД исходного запроса (в нижнем регистре)
error	Да	string	текст ошибки
comment	Нет	string	комментарий к ошибке: расширенное описание, уточнение причины или пути решения ошибки

#### ▼ Пример модели

```
{
  "requestId": "395f9f71-13c0-4b03-8228-c9c82f47a351",
  "error": "Количество выдаваемых УИД 4 ограничено относительно запрошенных 20 с учетом выданных ранее и неис",
  "comment": "Текст ошибки"
}
```

## 2. userError

Ошибка обработки из конкретного сервиса, **кроме интеграционной шины**. И данный объект подставляется в ответную модель данного сервиса в поле userError.

Название	Обязат.	Тип	Описание
traceId	Да	uuid	ИД трассировки в логгере
object	Да	int	объект ошибки
stage	Да	int	этап, на котором произошла ошибка
code	Да	int	код ошибки
message	Да	string	человеческое описание ошибки
comment	Нет	string	инструкция для исправления

### ▼ Пример модели

```
{
  "traceId": "395f9f71-13c0-4b03-8228-c9c82f47a351",
  "object": 1,
  "stage": 1,
  "code": 1,
  "message": "Документ. Проверка. Истек сертификат подписи. Код трассировки: 395f9f71-13c0-4b03-8228-c9c82f47",
  "comment": "Обновите сертификат подписи. Это возможно сделать на %реклама%"
}
```

## 7.5. Registration

Регистрация

Группа сообщений для работы с регистрацией клиентов и их сотрудников.

### 7.5.1. Описание командных моделей

Данный блок содержит описание командных моделей.

#### 1. createAbonentCommandEvent

Команда регистрации абонента.

Описание API информационной системы электронных перевозочных документов

Название	Обязат.	Тип	Описание
requestId	Да	uuid	Идентификатор запроса (в нижнем регистре)
abonentId	Да	uuid	ИД организации (участника документооборота) с префиксом оператора
abonentType	Нет	abonentType	Тип абонента (ПО)
inn	Да	string	ИНН организации/ИП
kpp	Нет	string	КПП организации
ogrn	Нет	string	ОГРН компании
taxAuthority	Нет	taxAuthority	Данные налогового органа ИФНС
organizationFullName	Да	string	Наименование организации полное
organizationShortName	Нет	string	Наименование организации сокращенное
additionalDetails	Нет	additionalDetails	Дополнительные реквизиты организации
legalAddress	Да	address	Юридический адрес организации
actualAddress	Нет	address	Фактический адрес организации
bankDetails	Нет	bankDetails	Банковские реквизиты организации
partnerInfo	Нет	partnerInfo	Информация по партнеру
dateRegistration	Да	string(date-time)	Дата создания заявки
phone	Нет	string	Контактный телефон организации
email	Нет	string	E-mail организации
tariff	Нет	tariff	Тариф
employees	Нет	employee[array]	Список сотрудников организации

## ▼ Пример модели

```
{
  "requestId": "a8b53313-4603-4988-a91b-b313d262a6a6",
  "abonentId": "b8b53313-4603-4988-a91b-b313d262a6a7",
  "abonentType": 11,
  "inn": "9616560990",
  "kpp": "999901001",
  "ogrn": "8134821960606",
  "taxAuthority": {
    "code": "9999",
    "name": "Межрайонная инспекция ФНС России № 99 по г. Тестостан"
  },
  "organizationFullName": "общество с ограниченной ответственностью Грузоотправитель",
  "organizationShortName": "Грузоотправитель",
  "additionalDetails": {
    "okved": "01.02.03",
    "okpo": "4634958066",
    "oktmo": "15456454646",
    "okato": "03 000 000 000",
    "okpd": "00.00.00.000",
    "okopf": "20101",
    "okfs": "59"
  },
  "legalAddress": {
    "postalIndex": "353901",
    "regionCode": "23",
    "regionName": "Тестовый регион",
    "area": "Краснодарский край",
    "city": "Новороссийск",
    "settlement": "Тестовый населённый пункт",
    "street": "Аршинцева ул",
    "house": "1",
    "housing": "А",
    "apartment": "10"
  }
}
```

```

    },
    "actualAddress": {
      "postalIndex": "353901",
      "regionCode": "23",
      "regionName": "Тестовый регион",
      "area": "Краснодарский край",
      "city": "Новороссийск",
      "settlement": "Тестовый населённый пункт",
      "street": "Аршинцева ул",
      "house": "1",
      "housing": "A",
      "apartment": "10"
    },
    },
    "bankDetails": {
      "name": "Банк получателя",
      "bic": "044525700",
      "correspondentAccount": "3010181020000000700",
      "paymentAccount": "40817810099910005465",
      "address": "353901, г Тестовый, Тестовый р-н, ул Тестовая, д 404 стр 4"
    },
    },
    "partnerInfo": {
      "code": 1,
      "name": "АО \"Тестовый оператор\" ОПС ЭДО",
      "phone": "8 (000) 000 00 00",
      "email": "test@test.ru"
    },
    },
    "dateRegistration": "2022-09-01T13:03:05.670Z",
    "phone": "+7000000000",
    "email": "test@test.ru",
    "tariff": {
      "id": "68b09b27-5701-47ed-8468-98886351473b",
      "name": "Промо",
      "duration": 30,
      "limit": 100
    },
    },
    "employees": [{
      "employeeId": "1dcfe779-bf1b-45d8-b64a-a551010e5c19",
      "name": "Константин",
      "surname": "Константинопольский",
      "patronymic": "Константинович",
      "snils": "184-895-799 60",
      "emailEmployee": "test@test.tt",
      "phoneEmployee": "+7000000000",
      "department": "Отдел разработки",
      "position": "Директор",
      "warrant": {
        "number": "1f2289b0-649c-4b67-87a3-0ac3c027d91b",
        "startDate": "2022-02-11T13:03:05.670Z",
        "endDate": "2022-02-11T13:03:05.670Z",
        "warrantStatus": "Active"
      }
    }
  ]
}

```

## 2. updateAbonentCommandEvent

Команда обновления данных абонента.

Название	Обязат.	Тип	Описание
requestId	Да	uuid	Идентификатор запроса (в нижнем регистре)
abonentId	Да	string	ИД организации (участника документооборота) с префиксом оператора
kpp	Нет	string	ИД организации (участника документооборота) с префиксом оператора
taxAuthority	Нет	taxAuthority	Данные налогового органа ИФНС
organizationFullName	Нет	string	Наименование организации полное
organizationShortName	Нет	string	Наименование организации сокращенное
additionalDetails	Нет	additionalDetails	Дополнительные реквизиты организации
legalAddress	Нет	address	Юридический адрес организации
actualAddress	Нет	address	Фактический адрес организации
bankDetails	Нет	bankDetails	Банковские реквизиты организации
phone	Нет	string	Контактный телефон организации
email	Нет	string	E-mail организации
tariff	Нет	tariff[array]	Тариф

## ▼ Пример модели

```

{
  "requestId": "a8b53313-4603-4988-a91b-b313d262a6a6",
  "abonentId": "2AE4E671C65-3DAB-4974-9FF9-3AEAF80AD723",
  "kpp": "999901001",
  "taxAuthority": {
    "code": "9999",
    "name": "Межрайонная инспекция ФНС России № 99 по г. Тестостан"
  },
  "organizationFullName": "общество с ограниченной ответственностью Грузоотправитель",
  "organizationShortName": "Грузоотправитель",
  "additionalDetails": {
    "okved": "01.02.03",
    "okpo": "4634958066",
    "oktmo": "15456454646",
    "okato": "03 000 000 000",
    "okpd": "00.00.00.000",
    "okopf": "20101",
    "okfs": "59"
  },
  "legalAddress": {
    "postalIndex": "353901",
    "regionCode": "23",
    "regionName": "Тестовый регион",
    "area": "Краснодарский край",
    "city": "Новороссийск",
    "settlement": "Тестовый населённый пункт",
    "street": "Аршинцева ул",
    "house": "1",
    "housing": "А",
    "apartment": "10"
  },
  "actualAddress": {
    "postalIndex": "353901",
    "regionCode": "23",
    "regionName": "Тестовый регион",
    "area": "Краснодарский край",
    "city": "Новороссийск",
    "settlement": "Тестовый населённый пункт",
    "street": "Аршинцева ул",
    "house": "1",
    "housing": "А",
    "apartment": "10"
  },
  "bankDetails": {
    "name": "Банк получателя",
    "bic": "044525700",
    "correspondentAccount": "3010181020000000700",
    "paymentAccount": "40817810099910005465",
    "address": "353901, г Тестовый, Тестовый р-н, ул Тестовая, д 404 стр 4"
  },
  "statusRegistration": "Accepted",
  "phone": "+70000000000",
  "email": "test@test.ru",
  "tariff": {
    "id": "68b09b27-5701-47ed-8468-98886351473b",
    "name": "Промо",
    "duration": 30,
    "limit": 100
  }
}

```

### 3. createEmployeeCommandEvent

Команда создания сотрудника абонента.

Название	Обязат.	Тип	Описание
requestId	Да	uuid	Идентификатор запроса (в нижнем регистре)
abonentId	Да	string	ИД организации (участника документооборота) с префиксом оператора
employee	Да	employee[array]	Сотрудник организации

#### ▼ Пример модели

```
{
  "requestId": "a8b53313-4603-4988-a91b-b313d262a6a6",
  "abonentId": "2AE4E671C65-3DAB-4974-9FF9-3AEAF80AD723",
  "employees": [{
    "employeeId": "1dcfe779-bf1b-45d8-b64a-a551010e5c19",
    "name": "Константин",
    "surname": "Константинопольский",
    "patronymic": "Константинович",
    "snils": "184-895-799 60",
    "emailEmployee": "test@test.tt",
    "phoneEmployee": "+70000000000",
    "department": "Отдел разработки",
    "position": "Директор",
    "warrant": {
      "number": "1f2289b0-649c-4b67-87a3-0ac3c027d91b",
      "startDate": "2022-02-11T13:03:05.670Z",
      "endDate": "2022-02-11T13:03:05.670Z",
      "warrantStatus": "Active"
    }
  }
]}
}
```

### 4. updateEmployeeCommandEvent

Команда обновления сотрудника.

Название	Обязат.	Тип	Описание
requestId	Да	uuid	Идентификатор запроса (в нижнем регистре)
employees	Да	employee[array]	Сотрудник организации

## ▼ Пример модели

```
{
  "requestId": "a8b53313-4603-4988-a91b-b313d262a6a6",
  "employees": [{
    "employeeId": "1dcfe779-bf1b-45d8-b64a-a551010e5c19",
    "name": "Константин",
    "surname": "Константинопольский",
    "patronymic": "Константинович",
    "snils": "184-895-799 60",
    "emailEmployee": "test@test.tt",
    "phoneEmployee": "+70000000000",
    "department": "Отдел разработки",
    "position": "Директор",
    "warrant": {
      "number": "1f2289b0-649c-4b67-87a3-0ac3c027d91b",
      "startDate": "2022-02-11T13:03:05.670Z",
      "endDate": "2022-02-11T13:03:05.670Z",
      "warrantStatus": "Active"
    }
  }]
}
```

## 5. addEmployeeCertificateCommandEvent

Команда добавления сертификата сотрудника организации.

Название	Обязат.	Тип	Описание
requestId	Да	uuid	Идентификатор запроса (в нижнем регистре)
employeeId	Да	uuid	Идентификатор сотрудника организации (в нижнем регистре)
certificateBody	Да	string(binary)	Файл открытого ключа сертификата в формате Base64

## ▼ Пример модели

```
{
  "requestId": "a8b53313-4603-4988-a91b-b313d262a6a6",
  "employeeId": "a9a53323-4621-4388-a94b-b3133234a6r4",
  "certificateBody": "ЭП в формате base64"
}
```

## 6. getRegistrationInfoCommantEvent

Команда для получения зарегистрированных данных абонента.

Название	Обязат.	Тип	Описание
requestId	Да	uuid	Идентификатор запроса (в нижнем регистре)
abonentId	Да	string	ИД организации (участника документооборота) с префиксом оператора

▼ Пример модели

```
{
  "requestId": "a8b53313-4603-4988-a91b-b313d262a6a6",
  "abonentId": "2AE4E671C65-3DAB-4974-9FF9-3AEAF80AD723"
}
```

## 7.5.2. Описание ответных моделей

### 1. responseStatusRequest

Сообщение о выполнении запроса.

Название	Обязат.	Тип	Описание
requestId	Да	uuid	Идентификатор запроса (в нижнем регистре)
status	Да	statusRequest	Код ответа
comment	Нет	string	Комментарий ошибки для запросов с ошибкой ( <code>status</code> не 200)

▼ Пример модели

```
{
  "requestId": "a8b53313-4603-4988-a91b-b313d262a6a6",
  "status": 200,
  "comment": "Запрос выполнен успешно"
}
```

## 2. abonentCreateOutboundEvent

Сообщение о выполнении запроса.

Название	Обязат.	Тип	Описание
requestId	Да	uuid	Идентификатор запроса (в нижнем регистре)
abonentId	Да	string	ИД организации (участника документооборота) с префиксом оператора
status	Да	statusRequest	Статус выполнения запроса
comment	Нет	string	Комментарий ошибки для запросов с ошибкой ( <code>status</code> не 200)

### ▼ Пример модели

```
{
  "requestId": "a8b53313-4603-4988-a91b-b313d262a6a6",
  "abonentId": "2AE4E671C65-3DAB-4974-9FF9-3AEAF80AD723",
  "status": 200
}
```

## 3. employeeCreateOutboundEvent

Сообщение о выполнении запроса.

Название	Обязат.	Тип	Описание
requestId	Да	uuid	Идентификатор запроса (в нижнем регистре)
employeeId	Да	uuid	Идентификатор сотрудника (в нижнем регистре)
status	Да	statusRequest	Статус выполнения запроса
comment	Нет	string	Комментарий ошибки для запросов с ошибкой ( <code>status</code> не 200)

### ▼ Пример модели

```
{
  "requestId": "a8b53313-4603-4988-a91b-b313d262a6a6",
  "employeeId": "b5676312-4603-4988-a91b-b313d262a6a6",
  "status": 200
}
```

## 4. abonentRegistrationInfoOutboundEvent

Сообщение о зарегистрированных данных абонента.

Название	Обязат.	Тип	Описание
requestId	Да	uuid	Идентификатор запроса (в нижнем регистре)
abonentId	Да	string	ИД организации (участника документооборота) с префиксом оператора
abonentType	Нет	abonentType	Тип абонента (ПО)
inn	Да	string	ИНН организации/ИП
kpp	Нет	string	КПП организации
ogrn	Да	string	ОГРН компании
taxAuthority	Да	taxAuthority	Данные налогового органа ИФНС
organizationFullName	Да	string	Наименование организации полное
organizationShortName	Нет	string	Наименование организации сокращенное
additionalDetails	Нет	additionalDetails	Дополнительные реквизиты организации
legalAddress	Да	address	Юридический адрес организации
actualAddress	Нет	address	Фактический адрес организации
bankDetails	Нет	bankDetails	Банковские реквизиты организации
partnerInfo	Нет	partnerInfo	Информация по партнеру
dateRegistration	Да	string(date-time)	Дата создания заявки
phone	Нет	string	Контактный телефон организации
email	Нет	string	E-mail организации
tariff	Нет	tariff	Тариф
employees	Нет	employee[array]	Список сотрудников организации

## ▼ Пример модели

```

{
  "requestId": "a8b53313-4603-4988-a91b-b313d262a6a6",
  "abonentId": "2AE4E671C65-3DAB-4974-9FF9-3AEAF80AD723",
  "abonentType": 11,
  "inn": "9616560990",
  "kpp": "999901001",
  "ogrn": "8134821960606",
  "taxAuthority": {
    "code": "9999",
    "name": "Межрайонная инспекция ФНС России № 99 по г. Тестостан"
  },
  "organizationFullName": "общество с ограниченной ответственностью Грузоотправитель",
  "organizationShortName": "Грузоотправитель",
  "additionalDetails": {
    "okved": "01.02.03",
    "okpo": "4634958066",
    "oktmo": "15456454646",
    "okato": "03 000 000 000",
    "okpd": "00.00.00.000",
    "okopf": "20101",
    "okfs": "59"
  },
  "legalAddress": {
    "postalIndex": "353901",
    "regionCode": "23",
    "regionName": "Тестовый регион",
    "area": "Краснодарский край",
    "city": "Новороссийск",
    "settlement": "Тестовый населённый пункт",
    "street": "Аршинцева ул",
    "house": "1",
    "housing": "А",
    "apartment": "10"
  },
  "actualAddress": {
    "postalIndex": "353901",
    "regionCode": "23",
    "regionName": "Тестовый регион",
    "area": "Краснодарский край",
    "city": "Новороссийск",
    "settlement": "Тестовый населённый пункт",
    "street": "Аршинцева ул",
    "house": "1",
    "housing": "А",
    "apartment": "10"
  },
  "bankDetails": {
    "name": "Банк получателя",
    "bic": "044525700",
    "correspondentAccount": "3010181020000000700",
    "paymentAccount": "40817810099910005465",
    "address": "353901, г Тестовый, Тестовый р-н, ул Тестовая, д 404 стр 4"
  },
  "partnerInfo": {
    "code": 1,
    "name": "АО \"Тестовый оператор\" ОПС ЭДО",
    "phone": "8 (000) 000 00 00",
    "email": "test@test.ru"
  }
}

```

```

"dateRegistration": "2022-09-01 00:00:00.000000",
"phone": "+70000000000",
"email": "test@test.ru",
"tariff": {
  "id": "68b09b27-5701-47ed-8468-98886351473b",
  "name": "Промо",
  "duration": 30,
  "limit": 100
},
"employees": [{
  "employeeId": "1dcfe779-bf1b-45d8-b64a-a551010e5c19",
  "name": "Константин",
  "surname": "Константинопольский",
  "patronymic": "Константинович",
  "snils": "184-895-799 60",
  "emailEmployee": "test@test.tt",
  "phoneEmployee": "+70000000000",
  "department": "Отдел разработки",
  "position": "Директор",
  "warrant": [{
    "number": "1f2289b0-649c-4b67-87a3-0ac3c027d91b",
    "startDate": "2022-02-11T13:03:05.670Z",
    "endDate": "2022-02-11T13:03:05.670Z",
    "warrantStatus": "Active"
  }]
}]
}

```

### 7.5.3. Описание дополнительных моделей

Данный блок содержит описание дополнительных моделей.

#### 1. taxAuthority

Информация о налоговом органе абонента.

Название	Обязат.	Тип	Описание
code	Да	string	Код налогового органа
name	Да	string	Наименование налогового органа

#### ▼ Пример модели

```

{
  "code": "9999",
  "name": "Межрайонная инспекция ФНС России № 2 по Республике Тестостан"
}

```

## 2. additionalDetails

Реквизиты организации.

Название	Обязат.	Тип	Описание
okved	Нет	string	Общероссийский Классификатор Видов Экономической Деятельности
okpo	Нет	string	Общероссийский Классификатор Предприятий и Организаций
oktmo	Нет	string	Общероссийский Классификатор Территорий Муниципальных Образований
okato	Нет	string	Общероссийский Классификатор Административно-Территориальных Образований
okpd	Нет	string	Общероссийский Классификатор Продукции по видам экономической Деятельности
okopf	Нет	string	Общероссийский Классификатор Организационно-Правовых Форм
okfs	Нет	string	Общероссийский Классификатор Форм Собственности

### ▼ Пример модели

```
{
  "okved": "01.02.03",
  "okpo": "4634958066",
  "oktmo": "15456454646",
  "okato": "0000000000",
  "okpd": "00.00.00.000",
  "okopf": "20101",
  "okfs": "59"
}
```

### 3. address

Адрес организации.

Название	Обязат.	Тип	Описание
postalIndex	Нет	string	Почтовый индекс
regionCode	Да	string	Код региона
regionName	Нет	string	Название региона
area	Нет	string	Район
city	Нет	string	Город
settlement	Нет	string	Населенный пункт
street	Нет	string	Улица
house	Нет	string	Дом
housing	Нет	string	Корпус
apartment	Нет	string	Квартира

▼ Пример модели

```
{
  "postalIndex": "000000",
  "regionCode": "000",
  "regionName": "Тестовый регион",
  "area": "Тестовый район",
  "city": "Тестовый город",
  "settlement": "Тестовый населённый пункт",
  "street": "Тестовая улица",
  "house": "404",
  "housing": "A",
  "apartment": "10"
}
```

#### 4. bankDetails

Банковские реквизиты организации/ИП.

Название	Обязат.	Тип	Описание
name	<b>Нет</b>	string	Наименование
bic	<b>Да</b>	string	Банковский идентификационный код
correspondentAccount	<b>Нет</b>	string	Корреспондентский счёт
paymentAccount	<b>Нет</b>	string	Расчетный счет
address	<b>Нет</b>	string	Адрес банка

▼ Пример модели

```
{
  "name": "Банк получателя",
  "bic": "044525700",
  "correspondentAccount": "30101810200000000700",
  "paymentAccount": "40817810099910005465",
  "address": "353901, г Тестовый, Тестовый р-н, ул Тестовая, д 404 стр 4"
}
```

#### 5. partnerInfo

Информация по партнеру.

Название	Обязат.	Тип	Описание
code	<b>Да</b>	integer(int32)	Код партнера
name	<b>Да</b>	string	Наименование партнера
phone	<b>Нет</b>	string	Номер телефона
email	<b>Нет</b>	string	Электронная почта

## ▼ Пример модели

```
{
  "code": 1,
  "name": "АО \"Тестовый оператор\" ОПС ЭДО",
  "phone": "8 (000) 000 00 00",
  "email": "test@test.ru"
}
```

## 6. tariff

Информация по партнеру.

Название	Обязат.	Тип	Описание
id	<b>Да</b>	integer(int32)	Идентификатор тарифа
name	<b>Да</b>	string	Наименование тарифа
duration	<b>Нет</b>	string	Срок действия тарифа в днях
limit	<b>Нет</b>	string	Лимит документов в тарифе

## ▼ Пример модели

```
{
  "id": "68b09b27-5701-47ed-8468-98886351473b",
  "name": "Промо",
  "duration": 30,
  "limit": 100
}
```

## 7. employee

Информация о сотруднике.

Название	Обязат.	Тип	Описание
employeeId	Да	uuid	Идентификатор сотрудника (в нижнем регистре)
name	Да	string	Имя
surname	Да	string	Фамилия
patronymic	Нет	string	Отчество
snils	Да	string	СНИЛС
emailEmployee	Нет	string	Email сотрудника
phoneEmployee	Нет	string	Номер телефона сотрудника
department	Нет	string	Подразделение
position	Нет	string	Должность
warrant	Нет	warrant[array]	Машиночитаемая доверенность (МЧД)
certificates	Нет	string[array]	Список сертификатов пользователя

## ▼ Пример модели

```
{
  "employeeId": "28029AF1-EB5C-42F8-8413-8B2FDD03607F",
  "name": "Константин",
  "surname": "Константинопольский",
  "patronymic": "Константинович",
  "snils": "184-895-799 60",
  "emailEmployee": "test@test.tt",
  "phoneEmployee": "+70000000000",
  "department": "Отдел разработки",
  "position": "Директор",
  "warrant": [{
    "number": "1f2289b0-649c-4b67-87a3-0ac3c027d91b",
    "startDate": "2022-02-11T13:03:05.670Z",
    "endDate": "2022-02-11T13:03:05.670Z",
    "warrantStatus": "Active"
  }],
  "certificates": [
    "ЭП в формате base64"
  ]
}
```

## 8. warrant

Информация о доверенности.

Название	Обязат.	Тип	Описание
number	Да	string	Номер доверенности
startDate	Да	string(date-time)	Дата начала действия доверенности
endDate	Да	string(date-time)	Дата окончания действия доверенности
warrantStatus	Да	warrantStatus	Статус доверенности

## ▼ Пример модели

```
{
  "number": "1f2289b0-649c-4b67-87a3-0ac3c027d91b",
  "startDate": "2022-02-11T13:03:05.670Z",
  "endDate": "2022-02-11T13:03:05.670Z",
  "warrantStatus": "Active"
}
```

## 9. warrantStatus

Статусы доверенности.

Название	Описание
Active	Действующая доверенность
Created	Доверенность существует, но не вступила в действие
Declined	Доверенность отозвана
Expired	Доверенность истекла

## 10. abonentType

Типы абонентов.

Название	Описание
11	Абоненты 1С Хаба
30	Абоненты
100	Неизвестный абонент

## 11. statusRequest

Коды ответов.

Код	Описание
200	Запрос выполнен успешно
400	Переданы некорректные параметры запроса
403	Отказано в доступе
500	Внутренняя ошибка сервера

### 7.6. Работа с приглашениями

Группа сообщений для работы с приглашениями.

### 7.6.1. Командные модели

Данный блок содержит описание основных командных моделей, отвечающих за выполнение действий.

#### 1. sendInvitation

Команда для отправки приглашения.

В случае когда нет параметра "recipientId", обязательно наличие параметров "inn" \ "kpp" для ЮР лиц или только "inn" для ИП.

Название	Обязат.	Тип	Описание
requestId	<b>Да</b>	uuid	Идентификатор запроса (в нижнем регистре)
senderId	<b>Да</b>	string	Идентификатор абонента с префиксом оператора
recipientId	<b>Нет</b>	string	Глобальный идентификатор контрагента, которому необходимо отправить приглашение
inn	<b>Нет</b>	string	ИНН получателя. Обязательно указывается если не указан recipientId
kpp	<b>Нет</b>	string	КПП получателя. Обязательно указывается если не указан recipientId
message	<b>Нет</b>	string	Сообщению получателю.

##### ▼ Пример модели #1

```
{
  "requestId": "00a7977d-a673-4080-9b93-1668790808eb",
  "senderId": "2AE0b75b312-796e-4bd9-a066-404cc015786a",
  "recipientId": "2AЕc1470d11-685b-4e2a-902f-69c31567127c"
}
```

##### ▼ Пример модели #2

```
{
  "requestId": "00a7977d-a673-4080-9b93-1668790808eb",
  "senderId": "2AE0b75b312-796e-4bd9-a066-404cc015786a",
  "inn": "4029017981",
  "kpp": "402901001",
  "message": "Давайте обмениваться документами ЭПД"
}
```

## ▼ Пример модели #3

```
{
  "requestId": "00a7977d-a673-4080-9b93-1668790808eb",
  "senderId": "2AE0b75b312-796e-4bd9-a066-404cc015786a",
  "inn": "770104294609",
  "message": "Давайте обмениваться документами ЭПД"
}
```

## 2. acceptInvitation

Команда для подтверждения приглашения.

Название	Обязат.	Тип	Описание
requestId	Да	uuid	Идентификатор запроса (в нижнем регистре)
abonentId	Да	string	Идентификатор абонента с префиксом оператора
invitationId	Да	int	Идентификатор приглашения

## ▼ Пример модели #1

```
{
  "requestId": "00a7977d-a673-4080-9b93-1668790809eb",
  "abonentId": "2AEc1470d11-685b-4e2a-902f-69c31567127c",
  "invitationId": "10001"
}
```

## 3. rejectInvitation

Команда для отклонения приглашения.

Название	Обязат.	Тип	Описание
requestId	Да	uuid	Идентификатор запроса (в нижнем регистре)
abonentId	Да	string	Идентификатор абонента с префиксом оператора
invitationId	Да	int	id приглашения

## ▼ Пример модели #1

```
{
  "requestId": "00a7977d-a673-4080-9b93-1658790809eb",
  "abonentId": "2AEc1470d11-685b-4e2a-902f-69c31567127c",
  "invitationId": "10002"
}
```

## 4. getIncomingInvitation

Команда для получения всех входящих приглашений для абонента.

Название	Обязат.	Тип	Описание
requestId	Да	uuid	Идентификатор запроса (в нижнем регистре)
abonentId	Да	string	Идентификатор абонента с префиксом оператора
begindate	Да	date	Дата от какого времени искать список входящих приглашений

## ▼ Пример модели #1

```
{
  "requestId": "00a7977d-a673-4080-9b93-1668790808eb",
  "abonentId": "2AEc1470d11-685b-4e2a-902f-69c31567128c",
  "begindate": "2006-01-02 15:04:05"
}
```

## 5. exportInvitationStatus

Команда для получения статуса приглашений.

Название	Обязат.	Тип	Описание
requestId	Да	uuid	Идентификатор запроса (в нижнем регистре)
invitationsId	Да	string [array]	id приглашений

## ▼ Пример модели #1

```
{
  "requestId": "00a7977d-a673-4080-9b93-1668790808eb",
  "invitationId": [
    "10001",
    "10002"
  ]
}
```

## 6. getAbonentId

Команда для получения ИД ЭДО абонента.

Название	Обязат.	Тип	Описание
requestId	<b>Да</b>	uuid	Идентификатор запроса (в нижнем регистре)
inn	<b>Да</b>	string	ИНН абонента
kpp	<b>Нет</b>	string	КПП абонента

## ▼ Пример модели #1

```
{
  "requestId": "00a7977d-a673-4080-9b93-1668790808eb",
  "inn": "4029017981",
  "kpp": "402901001"
}
```

## 7.6.2. Ответные модели

Данный блок содержит описание ответных моделей.

### 1. InvitationStatus

Параметры ответа.

Название	Обязат.	Тип	Описание
requestId	Да	uuid	Идентификатор запроса (в нижнем регистре)
invitations	Да	InvitationType[array]	Информация о приглашениях

#### ▼ Пример модели #1

```
{
  "requestId": "00a7977d-a673-5080-9b93-1658790809eb",
  "invitations": [{
    "Id": "10002",
    "Status": "sendInvitation",
    "senderId": "2AEc1470d11-685b-4e2a-902f-69c31567127c",
    "recipientId": "2AEc1470d11-685b-4e2a-902f-69c31567128c",
    "inn": "4029017981",
    "date": "2022-06-02 15:04:05"
  }]
}
```

### 2. responseAbonentId

Команда для получения ИДЭДО абонента.

Название	Обязат.	Тип	Описание
requestId	Да	uuid	Идентификатор запроса (в нижнем регистре)
abonentsId	Да	string [array]	Список идентификаторов абонента

## ▼ Пример модели #1

```
{
  "requestId": "00a7977d-a673-4080-9b93-1668790808eb",
  "abonentsId": [
    "2AE00a7977d-a673-4080-9b93-1668790808eb"
  ]
}
```

**7.6.3. Дополнительные модели**

Данный блок содержит описание дополнительных моделей.

**1. InvitationType**

Информация о приглашениях.

Название	Обязат.	Тип	Описание
Id	<b>Да</b>	int	id приглашения
Status	<b>Да</b>	InvitationStatus	Статус приглашения
senderId	<b>Да</b>	string	Идентификатор отправителя с префиксом оператора
recipientId	<b>Да</b>	string	Идентификатор получателя с префиксом оператора
inn	<b>Да</b>	string	ИНН отправителя
kpp	<b>Нет</b>	string	КПП отправителя
firstname	<b>Нет</b>	string	Имя отправителя приглашения (только для ИП)
lastname	<b>Нет</b>	string	Фамилия отправителя приглашения (только для ИП)
patronymic	<b>Нет</b>	string	Отчество отправителя приглашения (только для ИП)
date	<b>Да</b>	string	Дата поступления приглашения
message	<b>Нет</b>	string	Сообщению отправителя

## ▼ Пример модели #1

```

{
  "Id": "10002",
  "Status": "sendInvitation",
  "senderId": "2AEc1470d11-685b-4e2a-902f-69c31567127c",
  "recipientId": "2AEc1470d11-685b-4e2a-902f-69c31567128c",
  "inn": "4029017981",
  "kpp": "402901001",
  "firstname": "Иван",
  "lastname": "Иванов",
  "patronymic": "Иванович",
  "date": "2022-06-02 15:04:05",
  "message": "Давайте обмениваться документами ЭПД"
}

```

## 2. invitationStatus

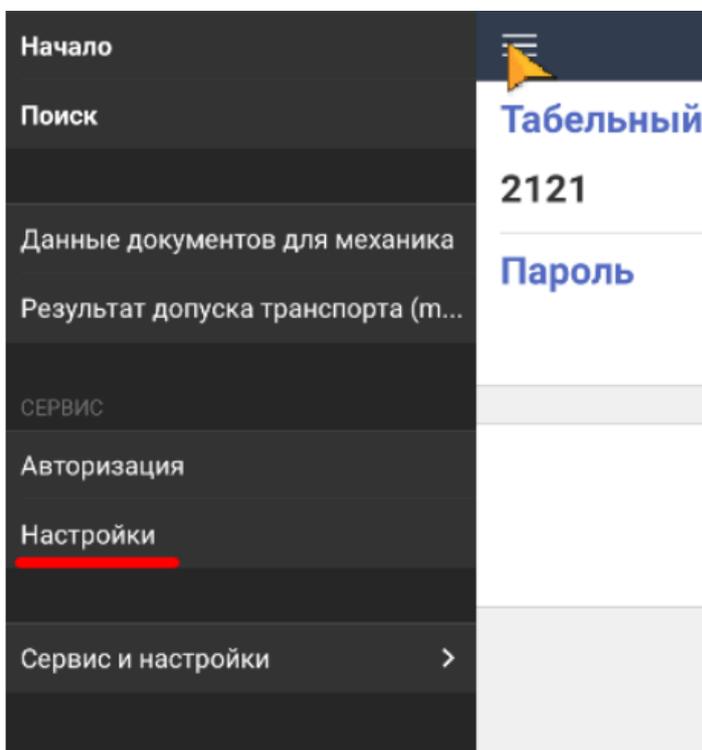
Типы статусов приглашений.

Название	Описание
sendInvitation	Отправлено приглашение контрагенту
acceptInvitation	Принято приглашение
rejectInvitation	Отклонено приглашение

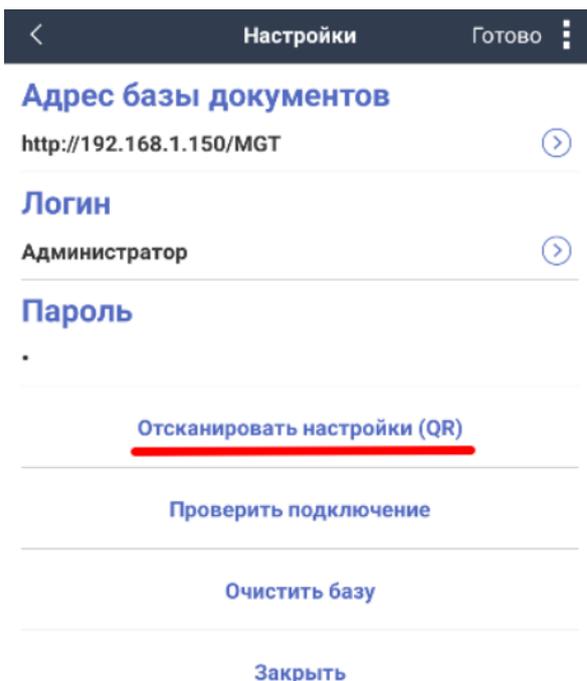
## 8. Приложение для водителей

### 8.1. Настройка

При первом запуске мобильного приложения необходимо указать настройки подключения к серверу. Для этого необходимо вернуться в главное меню и перейти в раздел «Настройки»:



Затем нажать кнопку «Отсканировать настройки (QR)», дать разрешение на доступ приложения к камере мобильного устройства и отсканировать QR - код (QR - код выдает администратор сервера либо иное уполномоченное лицо):



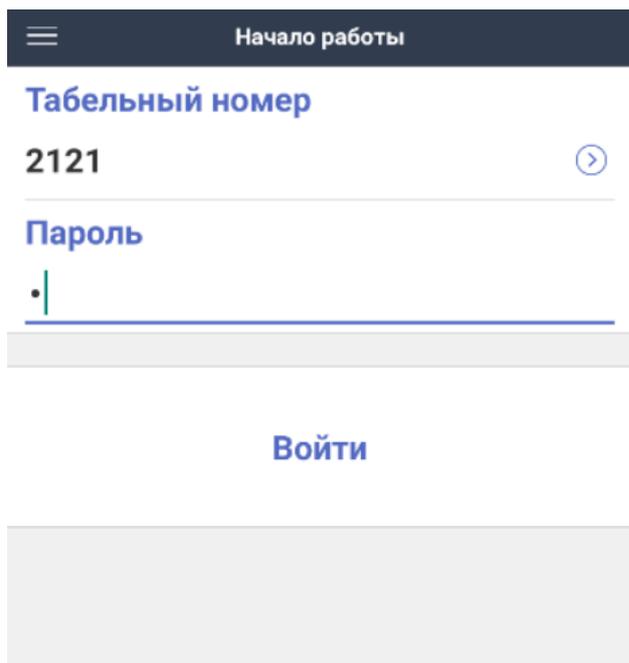
После того, как QR - код будет отсканирован, настройки автоматически пропишутся в приложении. Нажмите кнопку «Готово».

Мобильное приложение настроено и готово к работе.

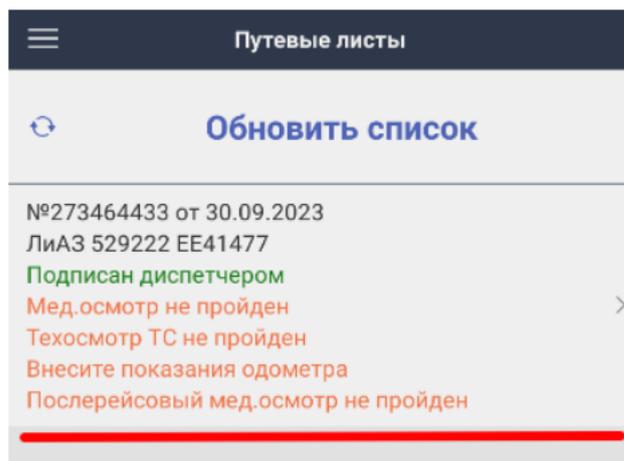
## 8.2. Авторизация

При запуске мобильного приложения открывается страница авторизации в приложение по логину (табельный номер сотрудника организации) и паролю пользователя (логин и пароль назначает и выдает администратор сервера либо иное уполномоченное лицо).

Введите табельный номер сотрудника и пароль, нажмите кнопку «Войти»:



После авторизации пользователь будет перенаправлен на страницу списка выписанных на него путевых листов:



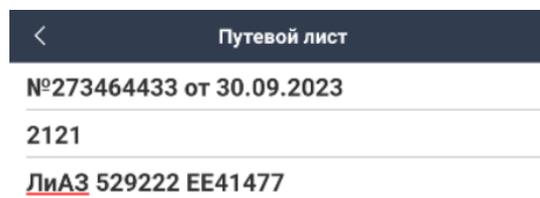
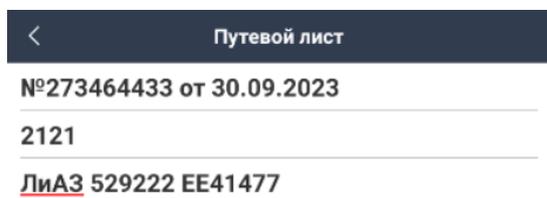
## 8.3. Прохождение предрейсового медицинского осмотра (подпись водителя)

После авторизации (входа) в приложении будет отображаться выписанный путевой лист.

После прохождения предрейсового медицинского осмотра водитель обязан поставить в приложении отметку о прохождении осмотра.

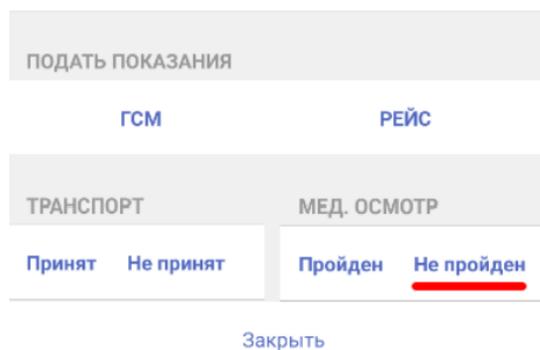
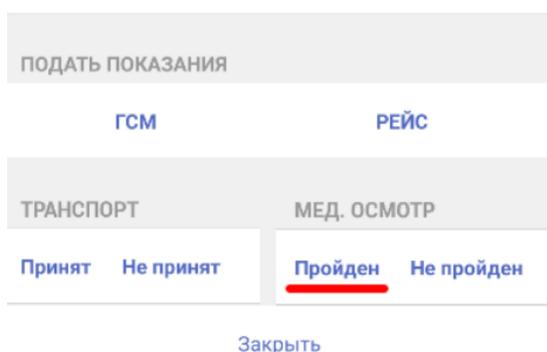
Для этого необходимо нажать на текущий путевой лист, перейти на страницу отображения данных путевого листа и внесения показаний.

Нажать на соответствующую кнопку «Пройден» или «Не пройден» в разделе «Мед. осмотр»:

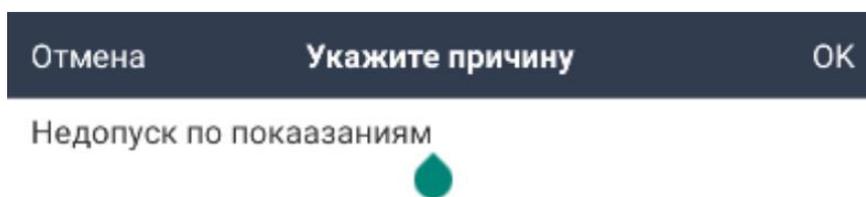


**Для формирования QR-кода требуется оформить титул внесения показаний одометра при выезде**

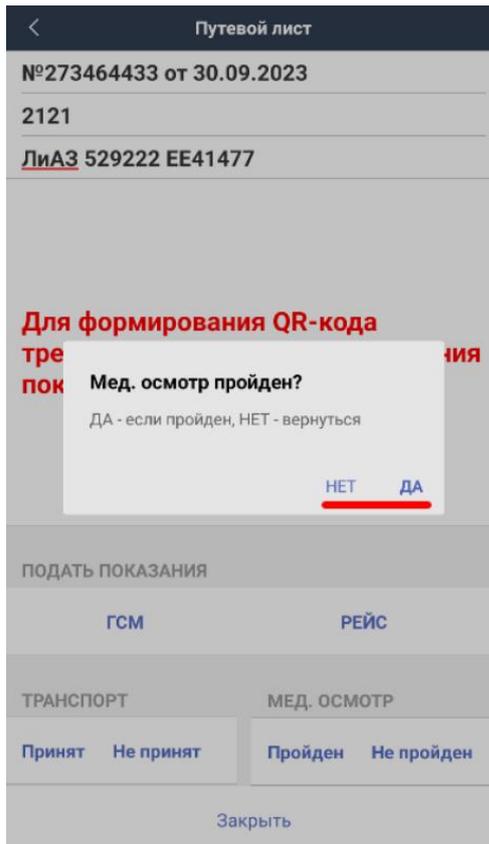
**Для формирования QR-кода требуется оформить титул внесения показаний одометра при выезде**



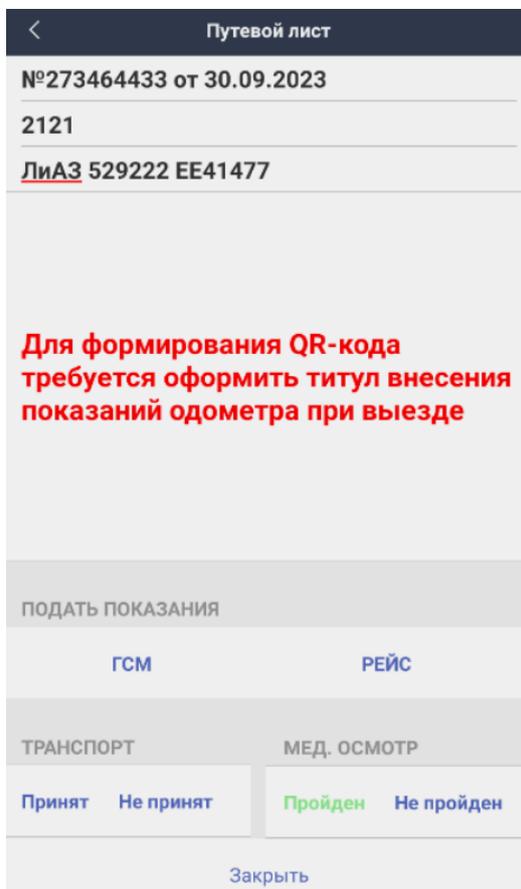
Если «Не пройден», то необходимо указать причину, например: «Недопуск по показаниям врача»:



Затем необходимо подтвердить свой выбор кнопками «Да» или «Нет»:



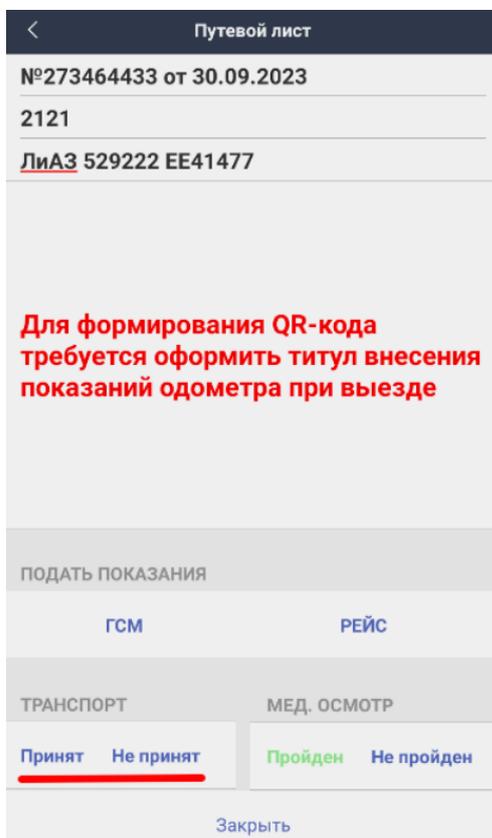
После подтверждения выбора в разделе «Мед. осмотр» кнопка изменит цвет на зеленый:



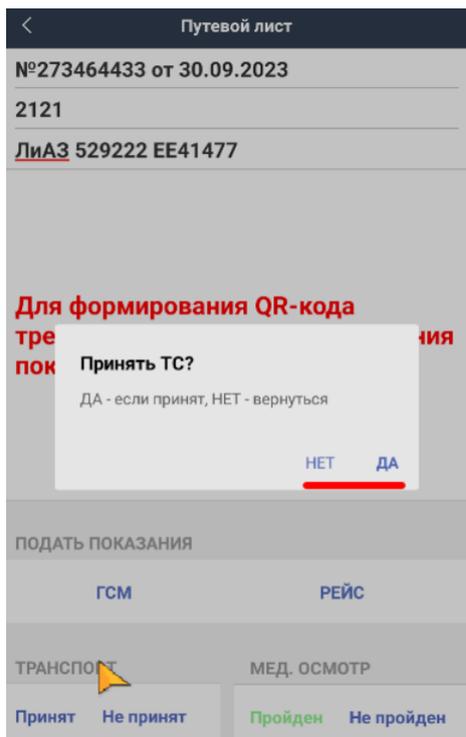
#### 8.4. Прием транспортного средства (подпись водителя)

При приеме транспортного средства водитель обязан поставить в приложении отметку о приеме либо об отказе в приеме транспортного средства.

Для этого необходимо нажать на текущий путевой лист и перейти на страницу отображения данных путевого листа и внесения показаний. В разделе «Транспорт» нажать соответствующую кнопку «Принят» или «Не принят»:



Затем необходимо подтвердить свой выбор кнопками «Да» или «Нет»:



После подтверждения выбора в разделе «Транспорт» кнопка изменит цвет на зеленый:

Путевой лист

№273464433 от 30.09.2023

2121

ЛиАЗ 529222 ЕЕ41477

**Для формирования QR-кода  
требуется оформить титул внесения  
показаний одометра при выезде**

ПОДАТЬ ПОКАЗАНИЯ

ГСМ РЕЙС

ТРАНСПОРТ МЕД. ОСМОТР

Принят Не принят Пройден Не пройден

Закреть

Если транспортное средство не принято, то необходимо указать причину, например: «Не работают фары»:

Отмена Укажите причину ОК

Не работают фары

## 8.5. Внесение показаний одометра при выезде/заезде

Перейдите в путевой лист и нажмите кнопку «Одометр»:

Заполните показания одометра на выезде и нажмите кнопку «Отправить показания»:

При заезде транспортного средства в парк повторите операцию с внесением показаний одометра в поле «Показания одометра на заезде»:

## 8.6. Внесение показаний остатка ГСМ при выезде/заезде

Перейдите в путевой лист и нажмите кнопку «ГСМ»:

Путевой лист

№273464433 от 30.09.2023

2121

ЛиАЗ 529222 ЕЕ41477

**Для формирования QR-кода  
требуется оформить титул внесения  
показаний одометра при выезде**

ПОДАТЬ ПОКАЗАНИЯ

ОДОМЕТР    **ГСМ**    РЕЙС

ТРАНСПОРТ    МЕД. ОСМОТР

Принят   Не принят    **Пройден**   Не пройден

Заккрыть

Заполните «Остаток топлива на выезде» и нажмите кнопку «Отправить показания»:

Показания ГСМ

Остаток топлива на выезде

**45**

Остаток топлива на заезде

**80**

**ЗАПРАВКИ**

Добавить заправку

**Отправить показания**

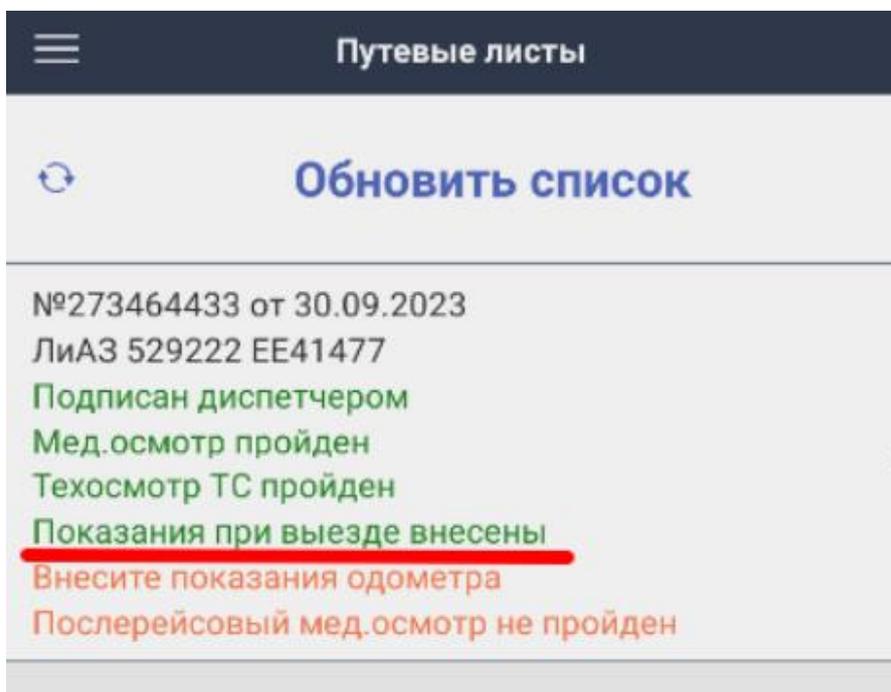
При заезде транспортного средства в парк повторите операцию с внесением показаний остатка топлива в поле «Остаток топлива на заезде»:



### 8.7. QR - код путевого листа

QR - код путевого листа необходим для предъявления сотрудникам ГИБДД по требованию.

QR - код доступен сразу после подписания путевого листа ответственным за выпуск ТС на линию. При этом статус путевого листа принимает значение «Показания при выезде внесены»:



Для предъявления QR - кода необходимо перейти в путевой лист из списка документов:

The screenshot displays a mobile application interface for a 'Путевой лист' (Waybill). At the top, there is a dark blue header with a back arrow and the text 'Путевой лист'. Below the header, the document number '№273464433 от 30.09.2023' is shown. Underneath, the number '2121' is displayed. The license plate 'ЛиАЗ 529222 ЕЕ41477' is also visible. A large QR code is centered on the screen. Below the QR code, there is a section titled 'ПОДАТЬ ПОКАЗАНИЯ' (Submit readings). This section contains three buttons: 'ОДОМЕТР' (Odometer), 'ГСМ' (GSM), and 'РЕЙС' (Flight). Below these buttons, there are two more sections: 'ТРАНСПОРТ' (Transport) and 'МЕД. ОСМОТР' (Medical examination). The 'ТРАНСПОРТ' section has two buttons: 'Принят' (Accepted) and 'Не принят' (Not accepted). The 'МЕД. ОСМОТР' section has two buttons: 'Пройден' (Passed) and 'Не пройден' (Not passed). At the bottom of the screen, there is a 'Закреть' (Close) button.

Если в пути нет возможности предъявить QR - код по некоторым причинам (закончился заряд мобильного устройства, поломка и т.д.), то инспектор имеет возможность просмотреть документ в своем личном кабинете информационной системы ГИС ЭПД по гос. номеру ТС.

## 8.8. Внесение показаний по кругорейсам

Для внесения показаний кругорейсов необходимо нажать кнопку «Рейс» и добавить рейс:

Путевой лист

№273464433 от 30.09.2023

2121

ЛиАЗ 529222 ЕЕ41477

ПОДАТЬ ПОКАЗАНИЯ

ОДОМЕТР    ГСМ    **РЕЙС**

ТРАНСПОРТ    МЕД. ОСМОТР

Принят   Не принят    Пройден   Не пройден

Закреть

Выбрать «Тип рейса» из списка возможных:

- Регулярный;
- Нулевой;
- Обед.

Указать плановое и фактическое время. Нажать кнопку «Готово»:

Готово

Тип рейса PP

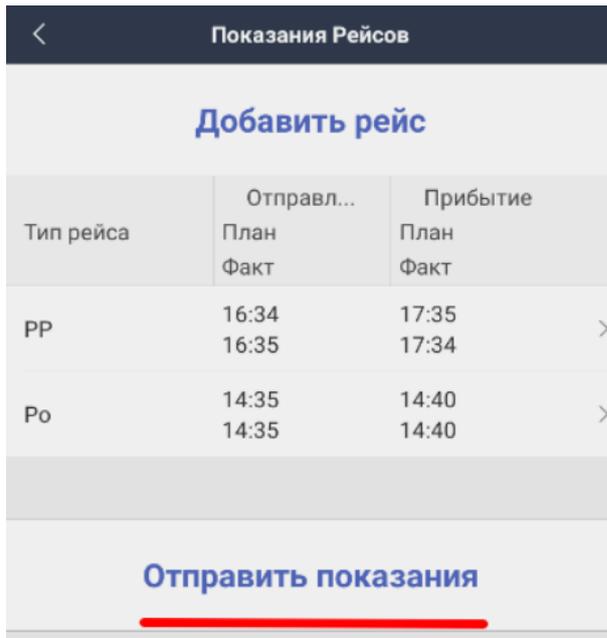
ОТПРАВЛЕНИЕ

План	16:34
Факт	16:35

ПРИБЫТИЕ

План	17:35
Факт	17:34

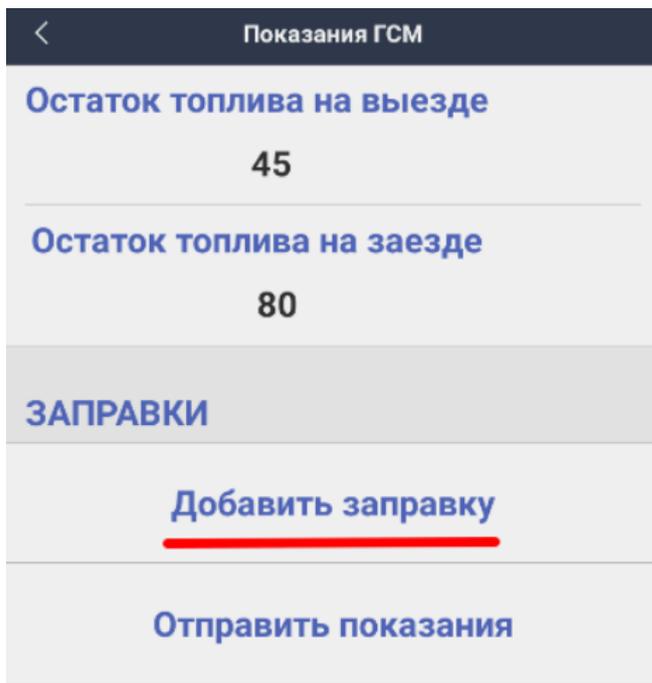
Затем нажать кнопку «Отправить показания»:



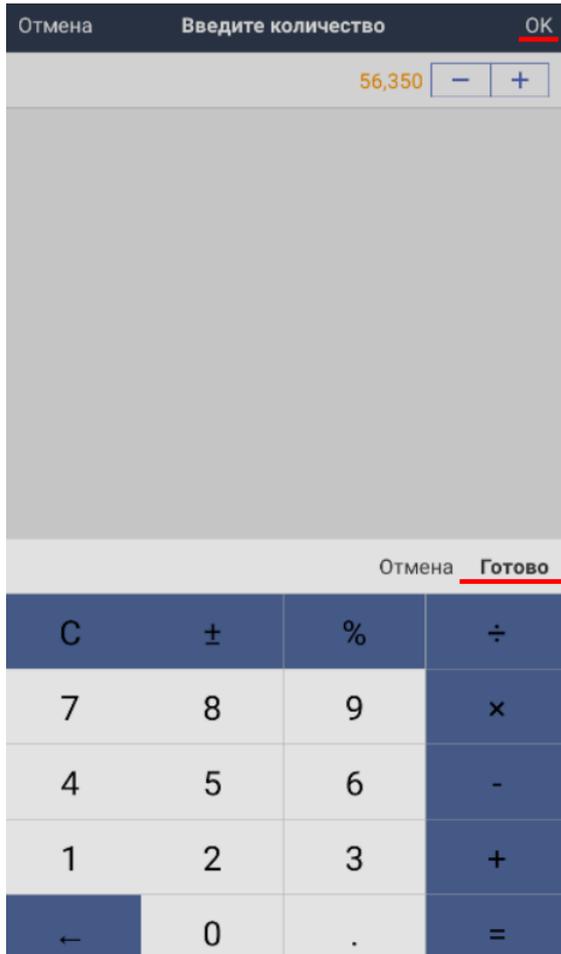
Отправить показания можно и один раз после указания всех типов рейса.

### 8.9. Внесение показаний по заправкам в течение рейса

Для внесения показаний заправки необходимо нажать кнопку «ГСМ» и «Добавить заправку»:



Указать количество литров ГСМ, нажать кнопку «Готово» и «Ок»:

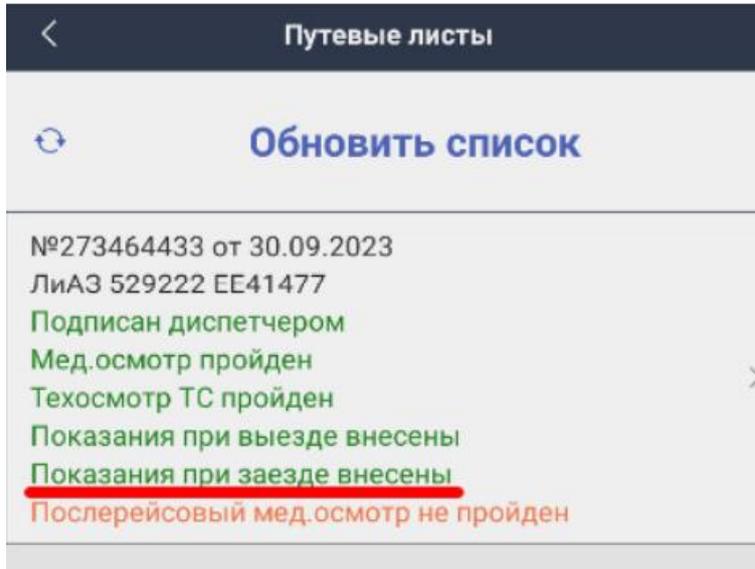


Затем нажать «Отправить показания»:



### 8.10. Послерейсовый медицинский осмотр

Когда ответственным за прием ТС при заезде в парк будет подписан путевой лист, статус путевого листа поменяется на «Показания при заезде внесены»:



Далее необходимо пройти послерейсовый медицинский осмотр.