

ООО «АКСИОМА»

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННО-РАСЧЕТНАЯ  
СИСТЕМА «АКСИОМА»**

**ИНСТРУКЦИЯ ПО ЭКСПЛУАТАЦИИ**

Версия документа: 1.0

На 205 листах

г. Санкт-Петербург

2026 г.

## ОГЛАВЛЕНИЕ

1	КРАТКОЕ ОПИСАНИЕ ПО.....	7
2	ОПИСАНИЕ ПО.....	8
2.1	Модуль «Интеграционный модуль эквайера для обслуживания ЕПГУ».....	8
2.1.1	Основные функции модуля ППС.....	8
2.1.2	Основные функции модуля РБ.....	9
2.1.3	Схема взаимодействия.....	11
2.1.4	ОПИСАНИЕ методов API для Модуля ППС.....	11
2.1.4.1	Оплата.....	11
2.1.4.1.1	Параметры Запроса на оплату .....	11
2.1.4.1.2	Параметры Ответа.....	16
2.1.4.1.3	Пример формирования поля browserdata.....	21
2.1.4.1.4	Пример формирования запроса на оплату.....	22
2.1.4.2	3ds-Аутентификация.....	24
2.1.4.2.1	Описание взаимодействия при 3ds-Аутентификации .....	24
2.1.4.2.2	Пример 3ds-Аутентификации .....	27
2.1.4.3	Статус платежа .....	28
2.1.4.3.1	Параметры запроса на статус платежа.....	28
2.1.4.3.2	Параметры ответа на статус платежа.....	29
2.1.4.3.2.1	Пример .....	29
2.1.4.4	Возврат платежа .....	30
2.1.4.4.1	Параметры Запроса на возврат платежа .....	30
2.1.4.4.2	Параметры Ответа на возврат платежа.....	31
2.1.5	Описание методов API для Модуля РБ.....	32
2.1.5.1	Указание на перечисление средств (ipsh:createPayOrder) .....	32
2.1.5.1.1	Запрос (ipsh:createPayOrderRequest).....	32
2.1.5.1.2	Ответ (ipsh:createPayOrderResponse) .....	36

2.1.5.2	Отмена платежа (ipsh:cancelPayment) .....	39
2.1.5.2.1	Запрос (ipsh:cancelPaymentRequest) .....	39
2.1.5.2.2	Ответ (ipsh:cancelPaymentResponse).....	39
2.1.5.3	Результат обработки отчета об исполнении платежей (ipsh:notifyRevisionResult) .....	40
2.1.5.3.1	Запрос (ipsh:notifyRevisionResultRequest).....	40
2.1.5.3.2	Ответ (ipsh:notifyRevisionResultResponse).....	40
2.1.5.4	Описание простых типов полей.....	40
2.1.5.5	Описание комплексных типов полей .....	45
2.1.5.6	Запрос к АБС о возможности проведения операции .....	54
2.1.5.6.1	Общие сведения .....	54
2.1.5.6.2	Подпись запроса.....	55
2.1.5.6.3	Формат запроса .....	55
2.1.5.6.4	Пример запроса: .....	56
2.1.5.7	Транзакционный отчет.....	57
2.1.5.7.1	Формат реестра.....	57
2.1.5.7.2	Заголовок реестра.....	57
2.1.5.7.3	Тело реестра.....	57
2.1.5.8	Платежный отчет.....	58
2.1.5.8.1	Формат реестра.....	58
2.1.5.8.2	Заголовок реестра.....	58
2.1.5.8.3	Тело реестра.....	58
2.1.6	Протокол взаимодействия .....	59
2.1.6.1	Общие сведения.....	59
2.1.6.2	Обмен сообщениями .....	59
2.1.6.3	Сжатие передаваемых данных .....	60
2.1.7	СПРАВОЧНИКИ.....	60
2.1.7.1	Тип распоряжения на оплату .....	60

2.1.7.2	Возможные типы дополнительных атрибутов .....	60
2.1.7.3	Возможные типы комиссий.....	61
2.1.7.4	Код причины отмены платежа .....	61
2.1.7.5	Статус распоряжения на оплату в расчетном банке .....	61
2.2	Модуль «Центр обработки транзакций».....	63
2.2.1	Протокол взаимодействия при оплате государственных услуг.....	63
2.2.2	Описание API.....	66
2.2.2.1	Метод search.....	66
2.2.2.2	Метод pay .....	76
2.2.2.3	Метод paymentInfo .....	110
2.2.2.4	Уведомление об успешном платеже notify .....	115
2.2.2.5	Метод kvit.....	116
2.3	Метод balance.....	117
2.3.1.1	Метод linkCard.....	119
2.3.1.2	Метод unlinkCard.....	120
2.3.1.3	Метод cards .....	121
2.3.1.4	Метод recurrent .....	123
2.3.1.5	Метод saveCard.....	126
2.3.1.6	Метод confirmSave .....	128
2.3.1.7	Метод searchPaid .....	129
2.3.1.8	Метод getPaymentOrde.....	132
2.3.1.9	Метод getInn.....	134
2.3.1.10	Уведомление о начислении по подписке subscribeNotify .....	136
2.3.1.11	Метод subscribe.....	139
2.3.1.12	Метод unsubscribe.....	141
2.3.1.13	Метод getSubscribeList.....	142
2.3.1.14	Метод photo.....	144
2.3.1.15	Метод getSBPBankList.....	145

2.3.1.16	Метод checkSBPPayee.....	147
2.3.1.17	Метод refund .....	149
2.3.1.18	Метод paymentinfoList .....	151
2.3.2	Описание типов данных .....	155
2.3.2.1	Ведомство (depType).....	155
2.3.2.2	Статусы Платежей (statuses) .....	156
2.3.2.3	Идентификаторы Пользователя (payerDoc).....	157
2.3.2.4	Коды ответов при возникновении ошибки при обработке запроса (error) .....	160
2.3.2.5	Данные по найденному Начислению (charge).....	163
2.3.2.6	Информация о Пользователе (payerParams) .....	167
2.3.2.7	Данные по Получателю платежа (paymentParams) .....	168
2.3.2.8	Пакет с реквизитами получателей для мультиоплаты авансовых платежей (packetAdvancePayments).....	170
2.3.2.9	Адрес пользователя для оплаты услуг ЖКХ (addressUserJkh) .....	172
2.3.2.10	Причины отмены платежа (cancelReason) .....	172
2.3.2.11	Криптограмма .....	173
2.3.2.12	Подпись (sign).....	174
2.3.3	Дополнительная информация по структурам и сценариям работы .....	180
2.3.3.1	Структура начислений и уведомлений .....	180
2.3.3.2	Алгоритмы привязки методов оплат .....	186
2.4	Модуль «Создания и ведения цифрового профиля пользователя» .....	193
2.4.1	Порядок взаимодействия систем .....	193
2.4.2	Структура полей JSON с данными о Клиенте .....	195
2.4.2.1	Описание наборов данных Клиента (UserData) .....	196
2.4.2.1.1	UserData для passport .....	196
2.4.2.1.2	UserData для passportExtDocCont .....	196
2.4.2.1.3	UserData для vehicleInfo.....	197

2.4.2.1.4	UserData для ProfileCont .....	198
2.4.2.1.5	UserData для ProfileContINN .....	199
2.4.2.1.6	UserData для PartnerProfile .....	199
2.4.2.1.7	UserData для PartnerProfileExtINN .....	200
2.4.2.2	Повторная отправка данных .....	201
2.4.2.3	Используемые коды ошибок .....	201
2.4.3	Примеры информационного обмена .....	202
2.4.3.1	Пример userData JSON для passport .....	202
2.4.3.2	Пример userData JSON для passportExtDocCont .....	202
2.4.3.3	Пример userData JSON для vehicleInfo .....	203
2.4.3.4	Пример userData JSON для ProfileCont .....	204
2.4.3.5	Пример userData JSON для ProfileContINN .....	204
2.4.3.6	Пример userData JSON для PartnerProfile .....	205
2.4.3.7	Пример userData JSON для PartnerProfileExtINN .....	205

# 1 КРАТКОЕ ОПИСАНИЕ ПО

Платформа «Аксиома» предназначена для обеспечения автоматизированного информационного и технологического взаимодействия между участниками платежных процессов при оплате государственных и муниципальных услуг, а также для обработки, передачи и предоставления данных в рамках интеграции с государственными и финансовыми системами.

Основной целью системы является оптимизация процесса оплаты и контроля задолженностей за счёт сокращения временных затрат пользователей, повышения прозрачности операций и минимизации необходимости личного посещения государственных учреждений и иных организаций.

Программное обеспечение реализовано по модели SaaS (Software as a Service), что позволяет пользователям получать доступ к функциональности сервиса через интернет без необходимости установки и обслуживания программного обеспечения на локальных устройствах. Все вычислительные процессы, обновления и хранение данных осуществляются на стороне поставщика сервиса, что обеспечивает актуальность системы, масштабируемость и высокую доступность. Использование SaaS-модели также упрощает интеграцию с внешними информационными системами и позволяет оперативно расширять перечень доступных услуг.

Программное обеспечение обеспечивает высокий уровень защиты персональных данных и финансовых операций за счёт использования современных технологий шифрования и соответствия действующим требованиям информационной безопасности.

## **2 ОПИСАНИЕ ПО**

### **2.1 Модуль «Интеграционный модуль эквайера для обслуживания ЕПГУ»**

Модуль состоит из следующих подмодулей:

1. Модуль ППС – предназначен для передачи в КО запроса на оплату начисления, а также получения от КО результата обработки запроса на оплату начисления;
2. Модуль РБ – предназначен для передачи в КО указания на перевод денежных средств, содержащее полный набор реквизитов, достаточный для того, чтобы сформировать платёжное поручение.

#### **2.1.1 Основные функции модуля ППС**

Одной из задач, решаемых модулем ППС, является взаимодействие с ИПШ, с целью передачи Поставщику платёжного сервиса запроса на оплату начисления, а также получения от Поставщика платёжного сервиса результата обработки запроса на оплату начисления.

Передача запроса на оплату Поставщику платёжного сервиса может быть выполнена путём:

1. перехода на платёжную страницу Поставщика платёжного сервиса с передачей необходимых данных через HTTPS запрос;
2. выставления счёта в систему Поставщика платёжного сервиса, который может быть оплачен пользователем позднее;
3. одновременного применения первого и второго действия;
4. передачи всех необходимых данных для выполнения списания в запросе на авторизацию без выполнения перехода на страницу платёжного сервиса;

Для разных типов Поставщиков платёжных сервисов (интернет-эквайринг, оплата через терминал, коммерческие платёжные системы, е-деньги, мобильные операторы и пр.) могут быть применены вышеперечисленные способы взаимодействия, приведённые к единому стандарту. Конкретный способ взаимодействия выбирается по согласованию с Поставщиком платёжного сервиса, учитывая особенности работы его информационной системы.

При выставлении счёта и переходе на платёжную страницу Поставщика платёжного сервиса ИПШ передаёт небольшой список параметров, достаточный для информирования пользователя о совершаемой операции. Если Поставщику платёжного сервиса необходимы дополнительные параметры платежа (например, для

печати чека), то он может обратиться к сервису получения полной информации по платежу.

Каждый Поставщик платёжного сервиса после проведения в его информационной системе оплаты (успешной, или неуспешной), должен сообщить результат оплаты ИПШ. По одному и тому же запросу на оплату может приходиться несколько уведомлений с результатом оплаты (например, первая попытка оплаты была неуспешной, а вторая попытка успешной).

Если при дальнейшей обработке платежа в ИПШ произошли какие-либо ошибки (например, Поставщик начислений сообщил о невозможности приёма уведомления о платеже, или Расчётный банк сообщил о невозможности перечисления денежных средств Поставщику начислений), то ИПШ направляет Поставщику платёжного сервиса запрос на отмену платежа. Поставщик платёжного сервиса должен выполнить отмену платежа, или вернуть фатальную ошибку, если отмена платежа невозможна. Отмена платежа выполняется в асинхронном режиме. После выполнения фактической отмены платежа Поставщик платёжного сервиса уведомляет ИПШ об изменении статуса платежа.

В запросе на оплату, передаваемом Поставщику платёжного сервиса, ИПШ указывает уникальный номер платежа в ИПШ. Соответственно в сообщении с результатом обработки платежа Поставщик платёжного сервиса должен указать тот же уникальный номер платежа в ИПШ.

Вместе с суммой платежа в запросе передаётся сумма комиссии, взимаемая с плательщика. Изменение суммы комиссии Поставщиком платёжного сервиса не допускается.

Взаимодействие с Операторами предпроцессинга осуществляется по тому же самому формату, что и взаимодействие с Поставщиками платёжных сервисов. При этом, поскольку предпроцессинг самостоятельно определяет банк-эквайер, то есть Поставщика платёжного сервиса, через которого будет выполнена оплата, то в уведомлении о проведении платежа предпроцессинг должен сообщить идентификатор Поставщика платёжного сервиса.

Для всех исходящих от ИПШ-СИСП сообщений время жизни сообщения установлено в константу, значение которой соответствует 20 секундам.

### **2.1.2 Основные функции модуля РБ**

Указания на перечисление денежных средств могут передаваться в расчётный банк в следующих случаях:

- Во время проведения платежа пользователем ИПШ формирует и отправляет в расчётный банк указание на перечисление суммы платежа Поставщику начислений

- С определённой регламентной периодичностью ИПШ формирует и отправляет в расчётный банк указания на перечисление суммы комиссии в пользу участников платёжной инфраструктуры, задействованных в проведении платежей
- Производится запрос в АБС для контроля возможности проведения платежа через ПС ЦБ.

ИПШ передаёт расчётным банкам указания на перевод денежных средств, содержащие полный набор реквизитов, достаточный для того, чтобы сформировать платёжное поручение.

В случае оплаты начислений Федерального казначейства, уведомление Федерального казначейства о произведённом платеже выполняют сами расчётные банки, поэтому ИПШ также передаёт в указании набор идентификаторов плательщика, необходимый для формирования уведомления Федерального казначейства о произведённом платеже.

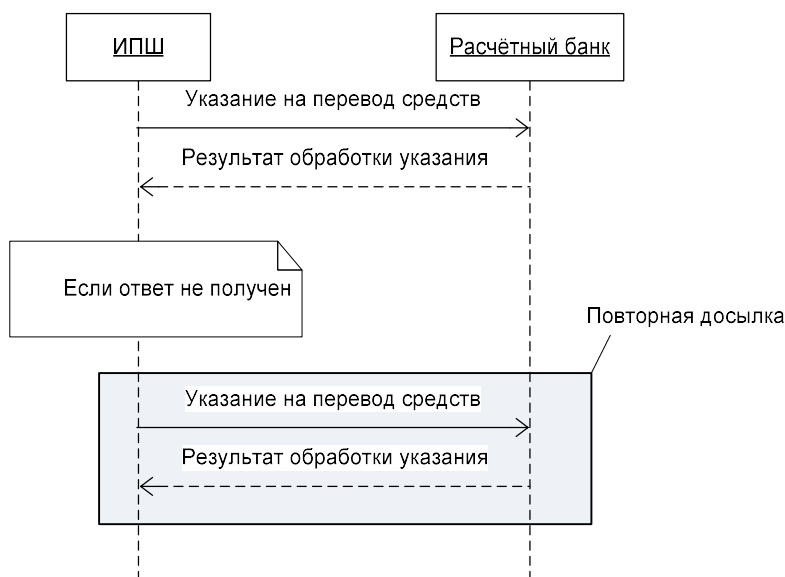
Если расчётный банк возвратил неуспешный результат обработки указания на перечисление денежных средств Поставщику начислений, то ИПШ обеспечивает полный откат транзакции по платежу – выполняет отмену платежа у Поставщика платёжного сервиса.

Если расчётный банк за определённый период времени не возвратил ответ с результатом обработки указания, то ИПШ считает, что с высокой вероятностью указание на перевод средств получено или будет успешно получено расчётным банком, и включает механизм повторной досылки указания в расчётный Банк.

После успешной обработки Расчётным банком указания на перечисление денежных средств ИПШ отправляет уведомление о факте платежа Поставщику начислений. В случае, если Поставщик начислений по каким-то причинам не принял уведомление о факте платежа (произошёл сбой при обработке уведомления о факте платежа у Поставщика начислений), то ИПШ отправляет Расчётному банку сообщение об отмене указания на перечисление денежных средств. Сообщение об отмене указания на перечисление денежных средств отправляется до 01:00:00 суток, следующих за датой и временем оплаты начисления пользователем. После 01:00:00 сообщения об отмене ИПШ в Расчётный банк не отправляет.

Формирование сверочных реестров в разрезах: платежей и транзакций. Основной причиной данного типа контроля обуславливается использование корзины платежей, что делает данную сверку обязательным инструментом.

### 2.1.3 Схема взаимодействия



### 2.1.4 Описание методов API для Модуля ППС

#### 2.1.4.1 Оплата

##### 2.1.4.1.1 Параметры Запроса на оплату

Параметры запроса				
URL	https://pay.***.ru			
метод	POST			
Параметры запроса	Поле	Описание (обязательные )	Формат (RegExp)	Примечания
	service	Код услуги	^\d{1,5}\$	Выдается Поставщику при заключении договора
	account	Основной идентификатор перевода	^.{1,128}\$	Номер телефона, договора, заказа и .т.п. По договорённости с Поставщиком может проверяться на уникальность.

amount	Сумма перевода	$\wedge \backslash d+(\backslash \backslash d\{1,2\})? \$$	При подключении рекуррентного перевода игнорируется
attr_1 .. n	Дополнительные атрибуты перевода	$\wedge .\{1,128\} \$$	Оговариваются заранее
sign	Подпись	$\wedge .\{1,128\} \$$	По согласованию, защита от подделки данных
CARD	PAN карты (*), либо строка APPLEPAY для платежа при помощи	$\wedge (\backslash d\{16,19\}) (APPLEPAY) (ANDROIDPAY) \$$	В запросе указываются либо эти данные, либо поле SOURCE. В случае указания в поле CARD значений APPLEPAY или ANDROIDPAY
EXP	Срок действия карты, месяц (*)	$\wedge \backslash d\{2\} \$$	обязательно передать дополнительную информацию в поле somedata
EXP_YEAR	Срок действия карты, год (*)	$\wedge \backslash d\{2\} \$$	
SOURCE	Перевод-донор (для услуг, поддерживающих режим упрощенного перевода)	$\wedge \backslash d\{1,9\} \$$	В запросе указывается либо это поле, либо должны быть переданы поля CARD, EXP, EXP_YEAR
CVC2	Контрольный код CVC2/CVV2 (*)	$\wedge \backslash d\{3\} \$$	
CARDHOLDER	Имя держателя карты,	$\wedge [0x20-0x5F]\{1,26\} \$$	

	указанное на ее лицевой стороне		
phone	Номер телефона	(^\d{10}\$)(^\+\d{9,15}\$)	Обязательно, если услуга оплачивается через веб-кошелек
email	Адрес электронной почты	^\.+@.+\$	
recurrent	Включение или выключение рекуррентного перевода	^((on) (off))\$	Если услуга поддерживает рекуррентные переводы, данный запрос включает такой режим для участвующей в нем карты и основного идентификатора перевода, либо выключает его. В случае передачи этого поля, поле amount передавать не надо.
rc_account	Основной идентификатор для рекуррентного перевода	^\.{1,128}\$	Необязательный параметр. Если передан, то рекуррентный перевод регистрируется для этого идентификатора, если нет, то для идентификатора, переданного в поле account
token	Токен для проведения рекуррентного перевода	^\d{1,9}\$	Если услуга поддерживает рекуррентные переводы, а также для данного основного

				идентификатора перевода этот режим включен и токен принадлежит этому идентификатору, то совершается рекуррентный перевод с использованием привязанной к токену карты
	subservice	Добавочный код услуги	^\d{1,9}\$	Необходим, если в рамках одной услуги необходимо организовать проведение авторизаций по разным терминалам, список добавочных кодов согласовывается с ПЦ/Поставщиком заранее
	ip	IP-адрес Плательщика	^\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\$	Если передан, то сохраняется в перевод, иначе в переводе сохраняется адрес, с которого пришел запрос
	client_account	Дополнительный идентификатор перевода	^\.{1,128}\$	В случае, если нет возможности передать идентификатор Плательщика в поле account (к примеру, account уникален), то привязка Плательщика к услуге (номер телефона, договора и т.п.) передается в этом поле.
	somedata	Дополнительные данные.	^\.{1,}\$	Зашифрованные по заранее обговорённому

	Обязательно при указании в качестве CARD значений APPLEPAY или ANDROIDPAY		алгоритму дополнительные данные перевода.  Правила заполнения для APPLEPAY и ANDROIDPAY указаны
hold	Признак предавторизации	^(1 0)\$	Если услуга поддерживает такой режим, то указывает, на двухфазную (1) или однофазную (0) схему обработки. Описание двухфазной схему смотри ниже.
ext_sess_id	Идентификатор внешней сессии	^.{1,128}\$	Идентификатор сессии платежа на стороне Клиента. Уникальность данного идентификатора не контролируется системой, но при использовании крайне желательно её соблюдать для дальнейшей проверки статуса платежа
browserdata	Упакованный в base64 объект, описывающий браузер пользователя	^[A-Za-z0-9+/=]{1,1024}\$	Обязателен для проведения 3DSecure 2.0 аутентификации. Если данное поле не будет передано или сформировано не корректно, то

				аутентификация по картам с поддержкой 3D Secure 2.0 будет проходить с ошибкой. Пример поля для передачи параметра описан ниже.
	xid	Поле XID из ответа MPI		При предварительном прохождении 3D Secure аутентификации на собственном MPI.
	cavv	Поле CAVV из ответа MPI		
	eci	Поле ECI из ответа MPI		

Поле sign вычисляется, если не оговорено другое, как md5 хэш от конкатенации следующих полей запроса или ответа (в указанном порядке):

service, account, amount, PaRes, PaReq, CReq, CRes, state и секретного ключа. Если какое-то поле не передается в запросе или ответе, то оно считается пустой строкой.

При получении запроса ПАК КО производит первичную проверку переданных данных.

После этого ПАК КО производит обработку всех переданных данных в ПЦ.

#### 2.1.4.1.2 Параметры Ответа

По окончании обработки транзакции ПАК КО возвращает ответ в виде JSON-массива:

Параметры ответа				
Параметр ответа	Поле	Описание (обязательные )	Формат (RegExp)	Примечания
	state	Код результата авторизации	^d{1,5}\$	
		<b>Значение</b>	<b>Расшифровка</b>	
		0	Перевод обработан успешно	

		1	Перевод находится в обработке, авторизация успешна	
		2	Перевод ожидает подтверждения одноразовым паролем	
		3	Для завершения привязки рекуррентного перевода необходимо передать сумму, заблокированную на карте Плательщика	
		-1	Транзакция отвергнута ПЦ, см. state_code	
		-2	Транзакция отвергнута ПАК Банка, см. state_code	
		-3	Неверные параметры перевода, перевод не прошел проверку у поставщика услуги	
		-4	Карта не привязана: возникает, если карта, с которой пытаются сделать оплату, не привязана к веб-кошельку или услуге, а это требуется, согласно настройке услуги	
		-5	Неизвестная ошибка, транзакция отвергнута	
		-8	Платеж просрочен, слишком много времени прошло между созданием платежа и вводом карточных данных	
		-12	Платёжный метод не поддерживается (для ApplePay, AndroidPay и т.п.)	
		-13	Двухфазные переводы не поддерживаются	
		-14	Не верная SMS подтверждения перевода для Веб-кошелька	
		-15	Рекуррентные переводы не поддерживаются	
		-16	Некорректные параметры для рекуррентного перевода, см. state_desc	
		-17	Подпись не верна	
		-18	Нарушение лимитов	
		-19	Предполагаемая попытка фрода	
	sign	Подпись	^.{1,128}\$	По согласованию, защита от подделки данных

state_code	Код ошибки	^[ -]? \d{1,5} \$		
	<b>Значение</b>	<b>Расшифровка</b>		
	<b>Если state=-2, ответы ПАК Банка</b>			
	100	Сервис недоступен		
	101	Регламентные работы		
	102	Недоступен шлюз в МПС		
	103	Технический сбой при обработке перевода, пользователь пытался задвоить транзакцию (нажал F5 в браузере)		
	104	Технический сбой при обработке перевода, разрушилась сессия на веб-сервере		
	105	Не прошла валидация полей		
	106	Не передан телефон (для услуги, оплачиваемой через веб-кошелек)		
	<b>Если state=-1, ответы МПС:</b>			
	-29	System error		
	-23	Invalid transaction context		
	-22	Invalid authentication information		
	-21	Duplicate transaction		
	-20	Expired transaction		
	-19	Authentication failed		
	-18	Error in CVC2 or CVC2 Description fields		
	-17	Access denied		
	-16	Terminal is locked, please try again		
	-15	Invalid Retrieval reference number		
	-14	PINpad agent/device error		
	-13	Unknown referer		
	-12	Error in merchant terminal field		
	-11	Error in currency field		
	-10	Error in amount field		
	-9	Error in card expiration date field		
	-8	Error in card number field		
	-7	Invalid response		
	-6	Configuration error		
	-5	Connect failed		

	-4	Server is not responding
	-3	No or Invalid response received
	-2	Bad CGI request
	00	Approved
	01	Call your bank
	03	Invalid merchant
	04	Your card is restricted
	05	Transaction declined
	06	Error - retry
	07	Your card is disabled
	9	Transaction is in processing...
	11	Declined
	12	Invalid transaction
	13	Invalid amount
	14	No such card
	15	No such card/issuer
	17	Declined
	19	Re-enter transaction
	20	Invalid response
	22	Declined
	30	Format error
	33	Expired card
	34	Suspected fraud
	35	Contact card issuer
	38	Allowable PIN tries exceeded
	41	Lost card
	43	Stolen card
	51	Not sufficient funds
	52	No chequing account
	53	No savings account
	54	Expired card
	55	Incorrect PIN
	57	Not permitted to client

		58	Not permitted to merchant	
		59	Suspected fraud	
		61	Exceeds amount limit	
		62	Restricted card	
		63	Security violation	
		65	Exceeds frequency limit	
		70	Reserved	
		72	Reserved	
		75	PIN tries exceeded	
		76	Wrong PIN,tries exceeded	
		77	No reference No.	
		78	Reserved	
		82	Time-out at issuer	
		86	Unable to verify PIN	
		88	Cryptographic failure	
		89	Authentication failure	
		90	Cutoff is in progress	
		91	Issuer unavailable	
		92	Router unavailable	
		93	Violation of law	
		94	Duplicate transmission	
		96	System malfunction	
		101	Merchant is not allowed recurring purchases	
		102	Token is not active	
		106	Merchant is not allowed preauth	
	state_desc	Текстовое описание ошибки	^.{1,128}\$	
	order	Номер квитанции	^\d{1,9}\$	Данные параметры могут не передаваться в случае, если авторизация транзакции не произошла
	account	Основной идентификатор перевода	^.{1,128}\$	

amount	Сумма перевода	^\d+(\.\d{1,2})?\$	
rc_sess_id	Сессия для рекуррентного перевода	^\d{1,9}\$	Только для рекуррентных переводов.
token	Токен рекуррентного перевода	^\d{1,9}\$	Данный токен ПЦ/ПОСТАВЩИК может в дальнейшем использовать для совершения операций без участия Плательщика. Возвращается только в случае успешной привязки.
pc	Код процессинга	^\w{1,9}^	Код процессинга, через который была авторизована транзакция
sign	Подпись	^\.{1,128}\$	По согласованию, защита от подделки данных

В случае ответа state=0 сумма списывается с карты Плательщика.

#### 2.1.4.1.3 Пример формирования поля browserdata

```

<form ...> ... <script language="JavaScript">
    var d = new Date();
    var userLang = navigator.language || navigator.userLanguage;
    var oBrowser = {
        brWidth:screen.width,
        brHeight:screen.height,
        brColorDepth:screen.pixelDepth,
        brTZ:d.getTimezoneOffset(),
        brLanguage:userLang,
        wWidth>window.innerWidth,
        wHeight>window.innerHeight,
        je:navigator.javaEnabled()
    };

```

```

document.write('<input type="text" name="browserdata" id="browserdata"
value="" + window.btoa(JSON.stringify(oBrowser)) + "" />');</script>
</script> ...
</form>

```

#### 2.1.4.1.4 Пример формирования запроса на оплату

Запрос:

```

CURL-Request:

POST / HTTP/1.1
Host: pay.***.ru
Accept: */*
Content-Length: 1616
Content-Type: application/x-www-form-urlencoded

array (
  'service' => '9732',
  'account' => 113975379,
  'amount' => '750.00',
  'sign' => '82b96aa4901fb62e446365fc731b0fa2',
  'somedata' =>
'g2T97xQ4eos7YJjW1xHOTjXq94KTPaE1QMW5rwjGfq9OUxuE1ywP9BJpMRZ1ex
A
/+mVav4IhtzRSW8BgDyYV1w==',
  'hold' => 0,
  'ext_sess_id' => '95f7dd57-8d72-4a39-ad5e-21f01c799f27',
  'browserdata' =>
'eyJicldpZHRoJjoxOTIwLCJickhlaWdodCI6OTc1LCJickNvbG9yRGVwdGgi
OjI0LCJiclRajoiLTI0MCIsImJyTGZuZ3VhZ2UiOiJydSIsIndXaWR0aCI6MTkyMCwi
d0hlaWdodCI6
MTA4MCwiamUiOiJ0cnVIn0=',
  'attr_BIC' => '017308101',

```

```

'attr_CBC' => '18811601121010001140',
'attr_CorrespAcc' => '40102810645370000061',
'attr_DrawerStatus' => '20',
'attr_KPP' => '732501001',
'attr_OKTMO' => '73701000',
'attr_PayeeINN' => '7303013280',
'attr_PersonalAcc' => '03100643000000016800',
'attr_Purpose' => 'УИН18810573241223001625///, Булатов Дамир Рустямович, №
188
10573241223001625 от 23.12.2024, Булатов Дамир Рустямович',
'attr_UrName' => 'УФК по Ульяновской области (УМВД России по Ульяновской
област
и)',
'attr_PayerINN' => '0',
'attr_PaytReason' => '0',
'attr_TaxPaytKind' => '0',
'attr_TaxPeriod' => '0',
'attr_Uin' => '18810573241223001625',
'attr_fio' => 'Булатов Дамир Рустямович',
)

```

ОТВЕТ:

```

{"MD":"f0291a6d-09c6-42f8-ba68-
d7c66a13fb8a","URL":"method","order":"122726873",
"threeDSServerTransID":"f0291a6d-09c6-42f8-ba68-d7c66a13fb8a","method_url":"http
s://3ds-
ds1.mirconnect.ru/ma","mresOk":"eyJ0aHJlZURTU2VydmVyVHJhbnNJRCI6ImYwMj
kx
YTZkLTA5YzYtNDJmOC1iYTY4LWQ3YzY2YTEzZmI4YSIsInRocmVIRFNDb21w
SW5kIjoiWSIsImJpIjoi
ZXIKaWNsZHBaSFJvSWpveE9USXdMQ0ppY2tobGFYZG9kQ0k2T1RjMUxDSmlja
052Ykc5eVJHVndkR2dp
T2pJMEExDSmljbFJhSWpvaUxUSTBNQ0lzSW1KeVRHRnVaM1ZoWjJVaU9pSnlkU0
lzSW5kWGFXUjBhQ0k2

```

```
TVRreU1Dd2lkMGhsYVdkb2RDSTZNVEE0TUN3aWftVWIPaUowY25WbEluMD0i
fQ=="mresFail":ey
J0aHJIZURTU2VydmVyVHJhbnNJRCI6ImYwMjkxYTZkLTA5YzYtNDJmOC1iYTY
4LWQ3YzY2YTEzZmI4YS
IsInRocmVIRFNDb21wSW5kIjoiTiIsImJpIjoiZXIKaWNsZHBaSFJvSWpveE9USXdM
Q0ppY2tobGFXZG
9kQ0k2T1RjMUxDSmlja052Ykc5eVJHVndkR2dpT2pJMEExDSmljbFJhSWpvaUxUST
BNQ0lzSW1KeVRHRn
VaM1ZoWjJVaU9pSnlkU0lzSW5kWGFXUjBhQ0k2TVRreU1Dd2lkMGhsYVdkb2RD
STZNVEE0TUN3aWftVW
lPaUowY25WbEluMD0ifQ=="sign":"6eee9ef3eb9bfe1646560a6c73aa18e9"}
```

## 2.1.4.2 3ds-Аутентификация

### 2.1.4.2.1 Описание взаимодействия при 3ds-Аутентификации

В случае необходимости перехода на сайт эмитента для подтверждения транзакции, ПАК КО возвращает ПЦ/ПОСТАВЩИКУ параметры данного перехода в виде JSON-массива.

Параметр ответа	Поле	Описание	Формат (RegExp)	Расшифровка
	sign	Подпись	^.{1,128}\$	По согласованию, защита от подделки данных
	PaReq	Поле запроса эмитенту		Данное поле надо передать в запросе к сайту эмитента
	MD	Поле запроса эмитенту		Данное поле надо передать в запросе к сайту эмитента
	URL	URL эмитента		Адрес, по которому должен быть отправлен Плательщик для подтверждения транзакции
	order	Номер квитанции	^d{1,10}\$	

В запросе к сайту эмитента ПЦ/ПОСТАВЩИК также должен указать поле TermUrl. После того, как Плательщик подтвердит транзакцию на сайте эмитента, эмитент произведет переход по адресу, указанному в этом поле. Переход будет выполнен в виде POST-запроса. ПЦ/ПОСТАВЩИК должен продублировать данный запрос для ПАК Банка для получения окончательного результата обработки транзакции.

Если карта плательщика поддерживает технологию 3D Secure 2.0, то может потребоваться дополнительно пройти шаг 3DS Method для передачи на аутентификацию данных об устройстве пользователя при помощи вызова в скрытом фрейме ACS эмитента.

Параметры данного перехода ПАК Банка возвращает ПЦ/ПОСТАВЩИКу в виде JSON-массива.

Поле	Описание	Формат (RegExp)	Расшифровка
sign	Подпись	^.{1,128}\$	По согласованию, защита от подделки данных
url	Признак необходимости прохождения шага 3DS Method	method	Ключевое слово.
MD	Поле запроса эмитенту		Данное поле надо передать в запросе к сайту эмитента
threeDSServerTransId	ID транзакции аутентификации	GUID	
threeDSMethodNotificationURL	URL ACS		Адрес ACS
order	Номер квитанции	^\d{1,10}\$	

mresOk	Код для возврата		
mresFail	Код для возврата		

Для передачи ACS данных об устройстве в скрытом фрейме ПЦ/ПОСТАВЩИК должен:

1. Сформировать поле threeDSMethodData как base64-закодированный json-объект с полями threeDSServerTransId из ответ ПАК КО, threeDSMethodNotificationURL – URL, на который ACS осуществит переход в случае успеха прохождения 3DS Method.
2. Выполнить передачу на ACS по адресу MethodURL поля threeDSMethodData в скрытом iframe в браузере Плательщика методом POST.
  - а. В случае успеха прохождения 3DS Method ACS вызовет threeDSMethodNotificationURL POST-запросом, передав в нём json-объект с полем threeDSMethodData = значение threeDSServerTransId.
  - б. Если в течении 10 секунд ACS не вызвал threeDSMethodNotificationURL или передал на него какие-то данные отличные, от threeDSServerTransId, 3DS Method считается не пройденным.

По результатам прохождения шага 3DS Method необходимо POST-запросом передать в ПАК КО поле mres (равное значению поля mresOk или mresFail из ответа на запрос проведения платежа) и поле TermURL. После того, как Плательщик подтвердит, при необходимости, транзакцию на сайте эмитента, эмитент произведет переход по адресу, указанному в этом поле.

При необходимости прохождения дополнительной аутентификации по протоколу 3DSecure 2.0 ПАК КО возвращает ПЦ/ПОСТАВЩИКу параметры данного перехода на сайт эмитента в виде JSON-массива.

Параметр ответа	Поле	Описание	Формат (RegExp)	Расшифровка
	sign	Подпись	^\{1,128\}\$	По согласованию, защита от подделки данных

	creq	Поле запроса эмитенту		Данное поле надо передать в запросе к сайту эмитента
	URL	URL эмитента		Адрес, по которому должен быть отправлен Платательщик для подтверждения транзакции
	order	Номер квитанции	^d{1,10}\$	

После того, как Платательщик подтвердит транзакцию на сайте эмитента, эмитент произведет переход по адресу, указанному в TermURL. Переход будет выполнен в виде POST-запроса. ПЦ/ПОСТАВЩИК должен продублировать данный запрос для ПАК КО для получения окончательного результата обработки транзакции.

#### 2.1.4.2.2 Пример 3ds-Аутентификации

```
{
  "TDS_DATA": "eyJ0aHJlZURTU2VydmVyVHJhbnNJRCI6ImYwMjkxYTZkLTA5YzYtNDJmOC1iYTY4LWQ3YzY2YTEzZmI4YSIsInRocmVIRFNNZXRob2R0b3RpZmljYXRpb25VUkwiOiJodHRwczpcL1wvM2RzLnRwcnMucnVcL25vdGlmeVwvNzA0MjQ0MzhLOWZjMzA5NDBiNzM3OGE1NWRiNzMyOGQifQ==",
  "ACSURL": "https://3ds-ds1.mirconnect.ru/vma"
}
```

```
array (
  'mres' => 'eyJ0aHJlZURTU2VydmVyVHJhbnNJRCI6ImYwMjkxYTZkLTA5YzYtNDJmOC1iYTY4LWQ3YzY2YTEzZmI4YSIsInRocmVIRFNDb21wSW5kIjoiWSIsImJpIjoiZXIKaWNsZHBaS FJvSWpveE9USXdM Q0ppY2tobGFYZG9kQ0k2T1RjMUxDSmlja052Ykc5eVJHVndkR2dpT2pJMExDSmlj bFJhSWpvaUxUSTBN Q0lzSW1KeVRHRnVaM1ZoWjJV aU9pSnlkU0lzSW5kWGFXUjBhQ0k2TVRreU1Dd
```

```

2lkMGhsYVdkb2RDSTZN
VEE0TUN3aWFtVWlPaUowY25WbEluMD0ifQ==',
'TermURL' => 'https://3ds.tprs.ru/redirect/70424438e9fc30940b7378a55db7328d',
)

```

### 2.1.4.3 Статус платежа

#### 2.1.4.3.1 Параметры запроса на статус платежа

Для проверки текущего состояния перевода ПЦ/ПОСТАВЩИК должна сделать HTTP-запрос к ПАК КО и обработать его ответ. Есть несколько вариантов запроса:

Параметры запроса				
URL	https://pay.***.ru/check/			
метод	POST			
Параметры запроса (вариант	Поле	Описание (обязательные)	Формат (RegExp)	Примечания
1)	service	Код услуги	^\d{1,5}\$	
	account	Основной идентификатор перевода	^\.{1,128}\$	
	amount	Сумма перевода	^\d+(\.\d{1,2})?\$	
	date	Дата перевода	^\d{2}\x2E\d{2}\x2E\d{4}\$	
	subservice	Добавочный код услуги	^\d{1,9}\$	
Параметры запроса (вариант 2)	service	Код услуги	^\d{1,5}\$	
	order	Номер квитанции	^\d{1,10}\$	Номер квитанции возвращается ПЦ/Поставщик в запросе на проведение перевода

	pin	4 последние цифры карты плательщика	^\d{4}\$	
	subservice	Добавочный код услуги	^\d{1,9}\$	
Параметр	service	Код услуги	^\d{1,5}\$	
ы запроса (вариант 3)	ext_sess_id	Внешний идентификато р сессии платежа	^\{1,128}\$	Идентификато р сессии, переданный на этапе создания платежа

### 2.1.4.3.2 Параметры ответа на статус платежа

ПАК КО возвращает ответ в виде JSON-массива

Элемент	Расшифровка	Формат	Примечание
order	Номер квитанции	^\d{1,2}\$	Равен нулю в случае какой-нибудь ошибки
state	Код ответа	^[ -]? \d{1,5}\$	
	Значение	Что означает	
	-7	Перевод отозван	
	-5	Другая ошибка	
	-2	Перевод не найден	
	-1	Перевод отвергнут	
	0	Перевод завершен	
	1	Перевод в обработке	

#### 2.1.4.3.2.1 Пример

Запрос:

array		(
'ext_sess_id'	=>	'95f7dd57-8d72-4a39-ad5e-21f01c799f27',

'service'	=>	'9732'
)		

Ответ 1:

```
{
  "account": "113975379",
  "amount": "750.00",
  "state": "1",
  "order": "122726873",
  "state_desc": "Платеж обрабатывается",
  "approval": "970645",
  "pc": "mc***_v2",
  "sign": "eb538f2c834d703d0c6f8fe93f2c6417"
}
```

Ответ 2:

```
{
  "account": "113975379",
  "amount": "750.00",
  "state": "0",
  "order": "122726873",
  "state_desc": "Платеж завершен",
  "approval": "970645",
  "pc": "mc***_v2",
  "sign": "9af6214fe0fd0488c4b0a249e47d7fb1"
}
```

#### 2.1.4.4 Возврат платежа

##### 2.1.4.4.1 Параметры Запроса на возврат платежа

Отзыв перевода возможен только по завершенным переводам. Отзыв возможен на сумму равную или меньшую, чем сумма первоначального перевода.

Для отзыва перевода ПЦ/ПОСТАВЩИК должен сделать HTTP-запрос к ПАК КО и обработать его ответ. Параметры запроса:

Параметры запроса				
URL	https://pay.***.ru			
метод	POST			
Параметры запроса	Поле	Описание (обязательные)	Формат (RegExp)	Примечания
	service	Код услуги	^\d{1,5}\$	
	account	Основной идентификатор отзываемого перевода	^\.{1,128}\$	

	amount	Сумма отзываемого перевода	$^{\backslash}d+(\backslash.d\{1,2\})?S$	Сумма должна быть равной или меньше суммы исходного перевода. Если сумма меньше суммы исходного перевода, то произойдет частичный возврат денежных средств на карту Плательщика
	reject	Номер квитанции отзываемого перевода	$^{\backslash}d\{1,9\}S$	Возвращается на Сайт ПЦ/ПОСТАВЩИК в сценарии оплаты
	confirm	4 правых цифры карты отзываемого перевода	$^{\backslash}d\{4\}S$	
	sign	Подпись	$^{\backslash}.\{1,128\}S$	По согласованию, защита от подделки данных

#### 2.1.4.4.2 Параметры Ответа на возврат платежа

ПАК КО возвращает ответ в виде JSON-массива с элементами state (код ответа) и state\_desc (тестовое описание ошибки, примечания). Код ответа может принимать следующие значения.

Код ответа	Расшифровка
-17	Подпись не верна
-7	Отзыв удался

-6	Перевод уже был отозван ранее
-5	Прочие ошибки
-2	Отзыв отвергнут, системная ошибка или не найден перевод
0	Вернуть средства на карту Плательщика невозможно
1	Отзыв находится в обработке
2	В настоящий момент отзыв перевода невозможен

## 2.1.5 Описание методов API для Модуля РБ

### 2.1.5.1 Указание на перечисление средств (ipsh:createPayOrder)

#### 2.1.5.1.1 Запрос (ipsh:createPayOrderRequest)

Описание параметров запроса «Указание на перечисление средств» приведено в расположенной ниже таблице.

Таблица 2. Описание параметров запроса «Указание на перечисление средств»

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
payOrder	Информация по отдельному указанию на перечисление средств	1..N	ipsh:PayOrderType
payOrderId	Идентификатор распоряжения на оплату	1..1	cn:Id
raceId	Идентификатор рейса, к которому относится платеж Рейс платежа определяет группу распоряжений об оплате платежей в течение дня в пользу отдельного ПН	1..1	cn:Id
payOrderTypeCode	Тип распоряжения на оплату	1..1	cn:PayOrderTypeCodeType
payOrderDate	Время отправки распоряжения на	1..1	xsd:dateTime

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
	оплату		
amount	Сумма, возвращенная по платежу	1..1	cn:AmountType
currencyCode	Код валюты платежа ("643")	1..1	cn:CurrencyCodeType
fee	Структура, содержащая информацию о комиссии, для которой выполняется перечисление. Для указаний на оплату платежей не передается.	0..1	ipsh:PayOrderFeeType
payment	Информация о платеже для оплаты которого было отправлено распоряжение на оплату. Для платежей по перечислению комиссии не возвращается	0..1	ipsh:PayOrderPaymentType
paymentId	Идентификатор платежа, по которому произошло возвращение средств	1..1	cn:Id
billNumber	Номер оплачиваемого начисления у ПН	1..1	cn:ExtIdType
paymentDate	Время платежа в ИПШ	1..1	xsd:dateTime
payMethodCode	Код способа платежа, с помощью которого выполняется оплата	1..1	cn:CodeType
paySystemCode	Код ППС через которую выполняется оплата	1..1	cn:CodeType

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
billProviderCode	Код ПН через которого выполняется оплата	1..1	cn:CodeType
serviceCode	Код услуги, за которую выполняется оплата	1..1	cn:CodeType
feeDetails	Структура с детализацией комиссий для платежа	1..1	cn:FeeDetailsType
fee	Сумма отдельной комиссии	1..N	cn:AmountType
fee/@feeCode	Код комиссии	1..1	cn:CodeType
transNumber	Код транзакции у ППС для обеспечения возможности сверки между ППС и расчетным банком	1..1	cn:ExtIdType
cardTransactionInfo	Структура для передачи данных по карточным платежам. Для других видов платежей не используется	0..1	cn:CardTransactionInfoType
grn	RRN	1..1	cn:String32Type
authCode	Код авторизации	1..1	cn:String32Type
payRequisites	Структура с реквизитами оплаты	1..1	cn:PayRequisitesType
receiverName	Наименование получателя	1..1	cn:String400Type
receiverInn	ИНН получателя	1..1	cn:String12Type

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
receiverKpp	КПП получателя	1..1	cn:String32Type
bic	БИК банка получателя	1..1	cn:String16Type
bankName	Наименование банка получателя	1..1	cn:String255Type
account	Счет получателя	1..1	cn:BankAccountNumberType
corrAccount	Корреспондентский счёт банка получателя	0..1	n:BankAccountNumberType
persAccount	Лицевой счет пользователя	0..1	cn:String20Type
payPurpose	Назначение платежа	1..1	cn:String255Type
okato	ОКАТО	0..1	cn:String32Type
kbk	КБК	0..1	cn:String64Type
payer	Структура информации плательщике. Присутствует при оплате платежей, данные плательщике необходимы для проведения таких платежей.	0..1	cn:PayerInfoType
payerName	Имя плательщика	1..1	cn:String255Type
payerMiddleName	Отчество плательщика	0..1	cn:String255Type
payerSurname	Фамилия плательщика	0..1	cn:String255Type
payerAddress	Адрес плательщика	1..1	cn:AddressType

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
zipCode	Почтовый индекс	0..1	xsd:string(6)
addressString	Адрес в строковом представлении	1..1	xsd:string(1000)
addrElements	Структурные элементы адреса	1..1	cn:AddrElements
addrElement	Элемент адреса	1..N	cn:AddrElement
level	Номер уровня элемента адреса	1..1	cn:Id
levelTypeID	Код типа уровня по справочнику типов уровней адресов	1..1	cn:Id
elmName	Наименование элемента адреса	1..1	cn:String255Type
elmCode	Код элемента по классификатору адресов	0..1	cn:String32Type
payerInn	ИНН плательщика	0..1	cn:String12Type
addAttributes	Дополнительные атрибуты платежа	0..1	cn:AddAttrsType
attr	Значение дополнительного атрибута	0..N	cn:AddAttrType
@name	Наименование дополнительного атрибута	0..1	cn:ExtIdType

#### 2.1.5.1.2 Ответ (ipsh:createPayOrderResponse)

Описание параметров ответа на запрос «Указание на перечисление средств» приведено в расположенной ниже таблице.

Таблица 3. Описание полей ответа на запрос «Указание на перечисление средств»

№	Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
	Error	Информация об ошибке обработки запроса	1..1	cn:Error
	errorCode	Код ошибки	0..1	cn:ErrorCode
	errorMessage	Описание ошибки	0..1	xs:string
	fieldErrors	Структура с описанием ошибок по валидации полей.	0..1	cn:FieldErrors
	fieldError	Поля с ошибками в запросе	0..N	cn:FieldError
	fieldName	Название поля с ошибкой.	1..1	xs:string
	error	Описание ошибки в запросе	1..1	cn:Error
	errorCode	Код ошибки	0..1	cn:ErrorCode
	errorMessage	Описание ошибки	0..1	xs:string
	addAttributes	Дополнительные атрибуты оплачиваемого начисления. Могут использоваться для передачи дополнительной информации ППС, РБ и ПН.	0..1	cn:AddAttrsType
	attr	Значение дополнительного атрибута	0..N	cn:AddAttrType

№	Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
	@name	Наименование дополнительного атрибута	0..1	xs:string
	payOrderErrors	Список ошибок, выявленных при обработке запроса.	0..N	ipsh:PayOrderErrorsType
	@payOrderId	Идентификатор распоряжения на оплату, к которому относятся ошибки	1..1	cn:Id
	payOrderError	Элементы содержат ошибки обработки распоряжений на оплату. Для одного распоряжения может быть возвращено несколько ошибок. При отсутствии ошибок в ответе считается, что все запросы на оплату приняты успешно. Если в ответе для отдельного распоряжения на оплату не возвращено ни одной ошибки, то считается, что прием данного распоряжения об оплате и проверка платежных реквизитов завершены	0..N	cn:CommonErrorType

№	Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
		успешно		
	errorCode	Код ошибки	1..1	cn:ExtIdType
	errorName	Краткое бизнес-наименование ошибки	1..1	cn:String255Type
	errorTime	Дата и время возникновения ошибки	1..1	xs:dateTime

## 2.1.5.2 Отмена платежа (ipsh:cancelPayment)

### 2.1.5.2.1 Запрос (ipsh:cancelPaymentRequest)

Описание параметров запроса на отмену платежа приведено в расположенной ниже таблице.

Таблица 1. Запрос на отмену платежа

Наименование элемента	Кол-во тэгов	Тип значения	Описание
paymentCancelInfo	1..N	cn:PaymentCancelInfoType	Структура для передачи запроса на отмену платежа. Элемент paymentCancelInfo/cancelTimeEnd не передается, т.к. в момент отправки запроса платеж не может быть окончательно отменен.

### 2.1.5.2.2 Ответ (ipsh:cancelPaymentResponse)

Описание параметров ответа на отмену платежа приведено в расположенной ниже таблице.

Таблица 5. Ответ на запрос отмены платежа

Наименование элемента	Кол-во тэгов	Тип значения	Описание
Error	1..1	cn:Error	Информация об ошибке, при обработке запроса
fieldErrors	0..1	cn:FieldErrors	Структура с описанием ошибок по валидации полей.
payOrderErrors	0..N	ipsh:PayOrderErrorsType	Содержит перечень возникших при приеме запроса ошибок, относящихся к отдельному платежу.
@payOrderId	1..1	xsd:long	Идентификатор платежа, к которому относятся ошибки

### 2.1.5.3 Результат обработки отчета об исполнении платежей (ipsh:notifyRevisionResult)

#### 2.1.5.3.1 Запрос (ipsh:notifyRevisionResultRequest)

Описание параметров запроса «Результат обработки отчета об исполнении платежей» приведено в расположенной ниже таблице.

Таблица 2. Параметры запроса «Результат обработки отчета об исполнении платежей»

Наименование элемента	Кол-во тэгов	Тип значения	Описание
revisionDate	1..1	xs:date	Дата, за которую выполняется сверка платежей
revisionTime	1..1	xs:dateTime	Дата и время выполнения запроса
raceId	1..1	cn:Id	Идентификатор рейса, к которому относятся платежи
payment	0..N	cn:RevisionPaymentResultType	Информация по отдельному платежу, по которому выявлено расхождение и выполнено корректирующее действие.

#### 2.1.5.3.2 Ответ (ipsh:notifyRevisionResultResponse)

Описание параметров ответа на запрос «Результат обработки отчета об исполнении платежей» приведено в расположенной ниже таблице.

Таблица 3. Параметры ответа на запрос «Результат обработки отчета об исполнении платежей»

Наименование элемента	Кол-во тэгов	Тип значения	Описание
Error	1..1	cn: Error	Информация об ошибке, при обработке запроса
fieldErrors	0..1	cn:FieldErrors	Структура с описанием ошибок по валидации полей.
payOrderErrors	0..N	ipsh: PayOrderErrorsType	Содержит перечень возникших при приеме запроса ошибок, относящихся к отдельному платежу.
@payOrderId	1..1	xsd:long	Идентификатор платежа, к которому относятся ошибки

#### 2.1.5.4 Описание простых типов полей

##### cn:Id

Общий тип для всех идентификаторов.

Базовый тип: xs:long.

**cn:ExtIdType**

Общий тип для всех идентификаторов внешних систем.

Базовый тип: cn:String64Type.

**cn:String6Type**

Общий тип для ограничения длинны строк в 6 символов.

Базовый тип: xs:string.

Значение должно быть не длиннее 6 символа(ов).

**cn:String12Type**

Общий тип для ограничения длинны строк в 12 символов.

Базовый тип: xs:string.

Значение должно быть не длиннее 12 символа(ов).

**cn:String16Type**

Общий тип для ограничения длинны строк в 16 символов.

Базовый тип: xs:string.

Значение должно быть не длиннее 16 символа(ов).

**cn:String20Type**

Общий тип для ограничения длинны строк в 20 символов.

Базовый тип: xs:string.

Значение должно быть не длиннее 20 символа(ов).

**cn:String32Type**

Общий тип для ограничения длинны строк в 32 символов.

Базовый тип: xs:string.

Значение должно быть не длиннее 32 символа(ов).

#### **cn:String64Type**

Общий тип для ограничения длинны строк в 64 символов.

Базовый тип: xs:string.

Значение должно быть не длиннее 64 символа(ов).

#### **cn:String255Type**

Общий тип для ограничения длинны строк в 255 символов

Базовый тип: xs:string

Значение должно быть не длиннее 255 символа(ов).

#### **cn:String400Type**

Общий тип для ограничения длинны строк в 400 символов

Базовый тип: xs:string

Значение должно быть не длиннее 400 символа(ов).

#### **cn:String1000Type**

Общий тип для ограничения длинны строк в 32 символов.

Базовый тип: xs:string.

Значение должно быть не длиннее 32 символа(ов).

#### **cn:String4000Type**

Общий тип для ограничения длинны строк в 4000 символов

Базовый тип: xs:string

Значение должно быть не длиннее 4000 символа(ов).

### **cn:CodeType**

Общий тип для всех кодов, за которые отвечает ИПШ.

Базовый тип: cn:String16Type.

### **cn:ExtIdType**

Общий тип для всех идентификаторов внешних систем.

Базовый тип: cn:String64Type.

### **cn:AmountType**

Тип для хранения суммы.

Базовый тип: xs:decimal.

Значение должно содержать не более 18 цифр и не более 2 цифр после запятой.

### **pay-bank:EnvType**

Тип для хранения кодов поставщиков для маршрутизации.

Базовый тип: xs:string.

Для типа установлены следующие ограничения на значения:  
"DEV", "UAT", "EXUAT", "SVCDEV", "TCOD", "PROD".

### **cn:CurrencyCodeType**

Тип для хранения кода валюты.

Базовый тип: xs:string.

Значение должно содержать 3 цифры.

### **cn:BankRevisionSolutionCodeType**

Базовый тип: xs:string.

Установленные ограничения в расположенной ниже таблице.

**Таблица 8. Ограничения на значения для типа cn:BankRevisionSolutionCodeType**

<b>Значение</b>	<b>Описание</b>
PAYMENT_CANCEL	Полная отмена платежа
BILL_SOURCE_PUSH	Допроведение платежа со стороны ИПШ путем уведомления ПН о платеже
PAY_BANK_PUSH	Допроведение платежа со стороны ИПШ путем передачи в РБ повторного распоряжения на перечисление средств
PAYMENT_STATUS_CHANGE	Изменение статуса платежа (приведение статуса платежа к статусу платежа в расчётном банке)
PAYMENT_STATUS_SAVE	Игнорирование статуса платежа (статус платежа не изменяется)
PAYMENT_IGNORE	Игнорирование платежа (поскольку платёж не найден в базе данных платежа, то он не обрабатывается)

### **cn:BankPayOrderStatusType**

Тип для хранения статусов распоряжения на оплату для расчетного банка.

Базовый тип: xs:string.

Установленные ограничения на значения указаны в расположенной ниже таблице.

**Таблица 4. Ограничения на значения для типа cn:BankPayOrderStatusType**

<b>Значение</b>	<b>Описание</b>
NEW	Распоряжение передано в банк, но не выполнено
CREATE_ERROR	При создании распоряжения возникла ошибка из-за валидации переданных параметров
CANCELED	Распоряжение отменено, согласно запросу на отмены платежа
PROCESSING_ERROR	При выполнении распоряжения возникла ошибка из-за параметров распоряжения
DONE	Распоряжение выполнено
MONEY BACK	По выполненному распоряжению были возвращены деньги
NOT_FOUND	Данные о платеже отсутствуют в полученной от банка сверке

## 2.1.5.5 Описание комплексных типов полей

Таблица 5. cn:AddAttrType

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
Attr	Значение дополнительного атрибута	1... N	cn:AddAttrType
@name	Наименование дополнительного атрибута	0..1	cn:ExtIidType, основан на cn:String255Type

Таблица 11. cn:FeeDetailsType

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
fee	Значение дополнительного атрибута	0.. N	cn: Fee_type0

Таблица 12. cn: Fee\_type0

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
@feeCode	Код типа комиссии	1..1	cn:CodeType, основан на cn:AmountType

Таблица 13. cn:CardTransactionInfoType

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
rrn	RRN	1..1	cn:String32Type
authCode	Код авторизации	0..1	cn:String32Type

Таблица 14. cn:PayRequisitesType

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
receiverName	Наименование получателя	1..1	cn:String400Type
receiverInn	ИНН получателя	1..1	cn:String12Type
receiverKpp	КПП	1..1	cn:String32Type

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
	получателя		
bic	БИК банка получателя	1..1	cn:String16Type
bankName	Наименование банка получателя	1..1	cn:String255Type
account	Счет получателя	1..1	cn:BankAccountNumberType
corrAccount	Корреспондентский счёт банка получателя	0..1	n:BankAccountNumberType
persAccount	Лицевой счет пользователя	0..1	cn:String20Type
payPurpose	Назначение платежа	1..1	cn:String255Type
okato	ОКАТО	0..1	cn:String32Type
kbk	КБК	0..1	cn:String64Type

Таблица 15. cn:PayerInfoType

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
payerName	Имя плательщика	1..1	cn:String255Type
payerMiddleName	Отчество плательщика	0..1	cn:String255Type
payerSurname	Фамилия плательщика	0..1	cn:String255Type
payerAddress	Адрес плательщика	1..1	cn:AddressType
payerInn	ИНН плательщика	0..1	cn:String12Type

Таблица 16. cn:AddressType

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
zipCode	Почтовый индекс	0..1	cn:String6Type

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
addressString	Адрес в строковом представлении	1..1	cn:String1000Type
addrElements	Структурные элементы адреса	1..1	cn:AddrElements

Таблица 17. cn:AddrElements

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
addrElement	Элемент адреса	1..N	cn:AddrElement

Таблица 6. cn:AddrElement

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
level	Элемент адреса	1..1	cn:Id
levelTypeID	Код типа уровня по справочнику типов уровней адресов	1..1	cn:Id
elmName	Наименование элемента адреса	1..1	cn:String255Type
elmCode	Код элемента по классификатору адресов	0..1	cn:String32Type

Таблица 19. cn:RevisionPaymentResultType

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
initState	Начальные статусы платежей в ИПШ и расчетном банке	1..1	cn:RevisionStatusCompareType
finalState	Конечные статусы платежей в ИПШ и расчетном банке по результату обработки реестра операций от расчетного банка	1..1	cn:RevisionStatusCompareType

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
solutionCode	Код результирующего действия, выполненного для согласования статусов	1..1	cn:BankRevisionSolutionCodeType

Тип основан на cn:RevisionPaymentResultType.

Таблица 20. cn:RevisionPaymentCommonType

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип	Комментарий
payOrderId	Начальные статусы платежей в ИПШ и расчетном банке	1..1	cn:Id	
paymentId	Конечные статусы платежей в ИПШ и расчетном банке по результату обработки реестра операций от расчетного банка	0..1	cn:Id	
amount	Код результирующего действия, выполненного для согласования статусов	1..1	cn:AmountType	
currencyCode	Код валюты платежа	1..1	cn:CurrencyCodeType	По умолчанию 643 (рубли)

Таблица 21. cn:CommonErrorType

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
errorCode	Код ошибки	1..1	cn:ExtIdType
errorName	Краткое бизнес-наименование ошибки	1..1	cn:String255Type
errorTime	Дата и время возникновения ошибки .	1..1	xs:dateTime
fatal	Признак фатальности ошибки.	0..1	xs:boolean
errorDetails	Техническая информация о произошедшей ошибке. Детальное описание причины ошибки,	0..1	cn:String4000Type

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
	информация об исключительной ситуации		
url	URL для передачи совместно с ошибкой. Может использоваться в отдельных типах ошибок	0..1	cn:String255Type

Таблица 22. cn:PaymentCancelInfoType

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип	Комментарий
paymentId	Уникальный идентификатор платежа в СИСП	1..1	xs:string	
userId	Идентификатор пользователя ЕПГУ, в следствии действий которого выполняется запрос	1..1	xs:string	
amount	Сумма отмены платежа	1..1	cn:AmountType	
fee	Сумма, на которую изменяется комиссия	1..1	cn:AmountType	
currencyCode	Валюта суммы оплаты («643»)	1..1	cn:CurrencyCodeType	
cancelReasonCode	Код причины отмены платежа.	1..1	cn:CancelReasonCodeType	
cancelTimeBegin	Время начала отмены платежа	1..1	xsd:dateTine	
cancelTimeEnd	Время окончания отмены платежа	0..1	xsd:dateTine	Заполняется только после окончания процесса отмены платежа. Используется в сервисе запроса деталей по платежу для предоставления информации.
cancelComment	Комментарий к причине отказа	1..1	cn:String255Type	

Таблица 23. cn:RevisionPaymentType

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
payOrderId	Идентификатор распоряжения на оплату	1..1	cn:Id
paymentId	Идентификатор платежа	0..1	cn:Id
amount	Сумма выполненного платежа	1..1	cn:AmountType
currencyCode	Код валюты платежа ("643")	1..1	cn:CurrencyCodeType
status	Статус распоряжения на оплату	1..1	cn:BankPayOrderStatusType
payOrder	Информация о платежном поручении, которым перечислены деньги.	0..1	cn:PayOrderType

Таблица 24. ipsh:PayOrderType

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
payOrderId	Идентификатор распоряжения на оплату	1..1	cn:Id (xsd:long)
raceId	Идентификатор рейса, к которому относится платеж Рейс платежа определяет группу распоряжений об оплате платежей в течение дня в пользу отдельного ПН	1..1	cn:Id (xsd:long)
payOrderTypeCode	Тип распоряжения на оплату.	1..1	cn:PayOrderTypeCodeType
payOrderDate	Время отправки распоряжения на оплату	1..1	xsd:dateTime
amount	Сумма, возвращенная по платежу	1..1	cn:AmountType
currencyCode	Код валюты платежа	1..1	cn:CurrencyCodeType
fee	Структура, содержащая информацию о комиссии, для которой выполняется перечисление.	0..1	ipsh:PayOrderFeeType

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
	Для указаний на оплату платежей не передается.		
payment	Информация о платеже для оплаты которого было отправлено распоряжение на оплату. Для платежей по перечислению комиссии не возвращается	0..1	ipsh:PayOrderPaymentType
payRequisites	Структура с реквизитами оплаты	1..1	cn:PayRequisitesType
payer	Структура с информацией о плательщике. Присутствует при оплате гос. платежей, т.к. данные о плательщике необходимы для проведения таких платежей.	0..1	cn:PayerInfoType
addAttributes	Дополнительные атрибуты платежа	0..1	cn:AddAttrsType

Таблица 257. ipsh:PayOrderPaymentType

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
paymentId	Идентификатор платежа, по которому произошло возвращение средств	1..1	cn:Id (xsd:long)
billNumber	Номер оплачиваемого начисления у ПН	1..1	xsd:string(64)
paymentDate	Время платежа в ИПШ	1..1	xsd:dateTime
payMethodCode	Код способа платежа, с помощью которого выполняется оплата	1..1	cn:CodeType
paySystemCode	Код ППС через которую выполняется оплата	1..1	cn:CodeType
billProviderCode	Код ПН через которого выполняется оплата	1..1	cn:CodeType
serviceCode	Код услуги, за которую выполняется оплата	1..1	cn:CodeType

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
feeDetails	Структура с детализацией комиссий для платежа	1..1	cn:FeeDetailsType
transNumber	Код транзакции у ППС для обеспечения возможности сверки между ППС и расчетным банком	1..1	cn:ExtIdType
cardTransactionInfo	Структура для передачи данных по карточным платежам. Для других видов платежей не используется	0..1	cn:CardTransactionInfoType

Таблица 26. ipsh:PayRequisitesType

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
receiverName	Наименование получателя	1..1	cn:String400Type
receiverInn	ИНН получателя	1..1	cn:String12Type
receiverKpp	КПП получателя	1..1	cn:String32Type
bic	БИК банка получателя	1..1	cn:String16Type
bankName	Наименование банка получателя	1..1	cn:String255Type
account	Счет получателя	1..1	cn:BankAccountNumberType
corrAccount	Корреспондентский счёт банка получателя	0..1	cn:BankAccountNumberType
persAccount	Лицевой счет пользователя	0..1	cn:String20Type
payPurpose	Назначение платежа	1..1	cn:String255Type

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
okato	ОКАТО	0..1	cn:String32Type
kbk	КБК	0..1	cn:String32Type

Таблица 27. ipsh:PayOrderErrorsType

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
@paymentId	Номер распоряжения на перечисление средств	0..1	cn:Id
payOrderError	Сообщения об ошибках, относящиеся к данному распоряжению	0..N	cn:CommonErrorType

Таблица 28. ipsh:PayOrderFeeType

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
feeCode	Код типа комиссии, по которому выполняется перечисление	0..1	cn:CodeType
paces	Структура с перечнем рейсов платежей, за которые выполняется перечисление комиссии. Для перепроверки суммы комиссии расчетному банку необходимо проверить сумму комиссии по платежам, которые выполнялись в рамках перечисленных в данной структуре рейсов загрузки.	0..1	Контейнер
raceId	Идентификатор отдельного рейса загрузки по которому будет выполнено перечисление комиссии	0..N	cn:Id

Таблица 29. pay-bank:PayBankServiceRequestType

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип	Комментарий
env	Код поставщика для маршрутизации	0..1	pay-bank:EnvType	По умолчанию «PROD»
routingCode	Код маршрутизации	1..1	xs:string	

Таблица 30. pay-bank: PayBankServiceResponseType

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип	Комментарий
env	Код поставщика для маршрутизации	0..1	pay-bank:EnvType	По умолчанию «PROD»

Таблица 31. cn:RevisionStatusCompareType

Код поля	Описание поля	Требования к заполнению	Способ заполнения/Тип
ipshState	Состояние распоряжения на оплату в ИПШ	0..1	cn:BankPayOrderStatusType
bankState	Состояние распоряжения на оплату в расчетном банке	0..1	cn:BankPayOrderStatusType

### 2.1.5.6 Запрос к АБС о возможности проведения операции

Протокол информационного обмена с АБС эквайера, запрос проверки соответствия реквизитов, для проведения платежей через систему срочных платежей ЦБ РФ.

#### 2.1.5.6.1 Общие сведения

После того, как все платежные поручения по каждому из платежей в пачке сформировались и отправились на обработку, платежный шлюз эквайера формирует и отправляет в ОУИО уведомление о принятых платежах.

Уведомление отправляется посредством HTTP-POST запроса. Данные передаются в виде json-объекта (формат см. ниже).

Для защиты канала необходимо использовать TLS-соединение. Желательна фильтрация запросов по IP-адресам на стороне оператора ОУИО.

Для аутентификации запросов можно использовать клиентский TLS-сертификат (если необходимо).

### 2.1.5.6.2 Подпись запроса

Для дополнительной защиты можно вычислять подпись запроса и передавать ее в соответствующем поле. В качестве подписи используется хэш SHA256 от строки, состоящей из параметров *split\_paymentId* + *secret*, где *secret* – секретная фраза. Для вычисления подписи нужно составить строку в формате UTF-8, полученную строку нужно преобразовать в массив байт и вычислить хэш.

### 2.1.5.6.3 Формат запроса

Для передачи данных уведомления используется json-объект со следующими полями:

Поле	Описание	Формат
state	Статус платежа. На данном этапе единственный статус – «Принят».	Строка. На данном этапе – константа «Accepted»
split_paymentId	Уникальный идентификатор платежа, переданный в соответствующем параметре в запросе при инициализации платежа.	Строка
items	Массив платежей	
AccDocNo	Номер платежного поручения	Строка
AccDocDate	Дата платежного поручения	Строка
Upno	Уникальный присваиваемый номер операции (УПНО/УИП)	Строка а формате УИП (УПНО)
PCID	Уникальные номер операции в процессинговом центре	Строка

orderID	Номер операции на портале ЕПГУ	Строка
sign	Подпись запроса. (см. Правила формирования подписи)	Строка

#### 2.1.5.6.4 Пример запроса:

```
{
  "state": "Accepted",
  "split_paymentId": "123213213",
  "items": [
    {
      "AccDocNo": "12345609",
      "AccDocDate": "27.07.2023",
      "Upno": "10440308520000032707202350000164",
      "PCID": "98798798",
      "orderID": "1"
    },
    {
      "AccDocNo": "12345608",
      "AccDocDate": "27.07.2023",
      "Upno": "10440308520000032707202350000165",
      "PCID": "98798798",
      "orderID": "2"
    },
    {
      "AccDocNo": "12345607",
      "AccDocDate": "27.07.2023",
      "Upno": "10440308520000032707202350000166",
      "PCID": "98798798",
      "orderID": "3"
    }
  ],
  "sign":
  "81d99ff10bf6b76c1fce30d7f30b19c709d0cae320f8aa30a6c7d6979d17e2b2"
}
```

## 2.1.5.7 Транзакционный отчет

### 2.1.5.7.1 Формат реестра

Имя файла: BANK\_OUIO\_trans\_DDMMYYYYY.csv

Формат CSV, кодировка UTF-8, разделитель «;»

Направляется ежедневно до 10:00 на support@\*\*\*.ru.

### 2.1.5.7.2 Заголовок реестра

Поле	Описание	Формат
Date	Дата реестра	Строка dd.mm.yyyyy
TotalAmount	Сумма списаний	Строка
QuantityPayments	Количество операций	Строка

### 2.1.5.7.3 Тело реестра

Поле	Описание	Формат
split_paymentId	Уникальный идентификатор платежа, переданный в соответствующем параметре в запросе при инициализации платежа.	Строка
paySystem	Платежная система	Строка
MCC	MCC код	Строка
card_number	Маскированный PAN	Строка
cardholder	Владелец карты	Строка
terminal_id	Номер терминала	Строка
dateTime	Дата и время транзакции	Строка
account	Основной идентификатор перевода	Строка
approval	Код авторизации	Строка
amount	Сумма списания	Строка
order	Номер квитанции	Строка
state	Статус платежа.	Строка
state_code	Код ошибки	Строка

## 2.1.5.8 Платежный отчет

### 2.1.5.8.1 Формат реестра

Имя файла: BANK\_OUIO\_DDMMYYYY.csv

Формат CSV, кодировка UTF-8, разделитель «;»

Направляется ежедневно до 10:00 на support@\*\*\*.ru.

### 2.1.5.8.2 Заголовок реестра

Поле	Описание	Формат
Date	Дата реестра	Строка dd.mm.yyyyy
QuantityPayments	Количество операций	Строка
TotalAmountAccepted	Сумма успешных списаний	Строка
TotalAmountToPayAccepted	Сумма успешных операций	Строка
TotalAmount	Сумма списаний	Строка

### 2.1.5.8.3 Тело реестра

Поле	Описание	Формат
ticket	Номер квитанции в ИЭ	Строка
payment_id	Уникальный номер платежа в ИС ПСКБ	Строка
AccDocNo	Номер платежного поручения	Строка
AccDocDate	Дата платежного поручения	Строка
Urno	Уникальный присваиваемый номер операции (УПНО/УИП)	Строка
PCID	Уникальные номер операции в процессинговом центре	Строка
orderID	Номер операции на портале ЕПГУ	Строка
terminal_id	Номер терминала	Строка
card_number	Маскированный PAN	Строка
amount	сумма списания операции для мультиплатежа	Строка
amountToPay	сумма операции для мультиплатежа	Строка

split_paymentId	Уникальный идентификатор платежа, переданный в соответствующем параметре в запросе при инициализации платежа.	Строка
state	Статус платежа.	Строка. «Accepted», «Rejected», «AcceptedNotNotified»

## 2.1.6 Протокол взаимодействия

### 2.1.6.1 Общие сведения

Модули реализованы с использованием клиент-серверной архитектуры.

Обмен информацией между Клиентом и Сервером выполняется с использованием запросов (Request) и ответов (Response). Начало процесса обмена информацией может быть инициировано только со стороны Клиента.

В качестве транспортного протокола используются протоколы HTTP/HTTPS.

Протокол работает в синхронном режиме. В ответе возвращается результат выполнения операции.

Протокол не использует пакетный режим. Это означает, что при отправке одного запроса выполняется только одна операция.

Обращение Клиента к Серверу выполняется посредством отправки HTTP-запроса, в теле которого передается сообщение, в формате json. Сервер должен отвечать Клиенту аналогичным образом (сообщением, в формате json).

### 2.1.6.2 Обмен сообщениями

Все запросы передаются по протоколу HTTP (версия 1.1, RFC 2616) методом POST. Сообщения передаются в теле HTTP-запроса.

Протокол не поддерживает разделенные (chunked) данные. Длина передаваемого на сервер запроса ограничена. По умолчанию ограничение составляет 10000 байт. Это значение может быть изменено администратором.

В соответствии с протоколом сообщения могут быть переданы на сервер в любой кодировке на усмотрение разработчика. В случае отсутствия явного указания кодировки в запросе или неверного названия кодировки, используется кодировка, установленная на платежном сервере по умолчанию. Ответ от сервера передается в той же кодировке, что была использована в запросе.

Для запросов и ответов должна использоваться кодировка UTF-8.

### 2.1.6.3 Сжатие передаваемых данных

Перед отправкой сообщений по сети клиент может сжать данные по алгоритму gzip или deflate. Информация об использованном методе сжатия должна передаваться в заголовке Content-Encoding. Если серверу не удастся произвести декомпрессию принятых данных или если было указано неверное значение заголовка, то сервер отвечает ошибкой 415.

Клиент может управлять сжатием ответов сервера. При этом при формировании ответа на Сервере используются те же алгоритмы, что были использованы для сжатия отправленного на Сервер запроса.

Выбранный метод передается в заголовке Accept-Encoding. В случае успешной обработки этого заголовка, Сервер указывает алгоритм сжатия ответа в заголовке Content-Encoding. Если же в запросе указан алгоритм, который не поддерживается, то сервер ответит ошибкой 406 (Not Acceptable).

## 2.1.7 СПРАВОЧНИКИ

### 2.1.7.1 Тип распоряжения на оплату

Таблица 32. Возможные значения для типа «Тип распоряжения на оплату»

Код	Название
PAYMENT	Распоряжение на перечисление денежных средств по платежу
FEE	Распоряжение на перечисление комиссий

### 2.1.7.2 Возможные типы дополнительных атрибутов

Таблица 8. Возможные типы дополнительных атрибутов

Тип атрибута	Тип значения	Описание
UnifiedPayerIdentifier	Текст	Единый идентификатор плательщика ФК
AltPayerIdentifier	Текст	Альтернативный идентификатор плательщика ФК
FKPurpose	Текст	Назначение платежа для налоговых платежей
FKTaxPeriod	Текст	Налоговый период для налоговых платежей
FKTaxDocNumber	Текст	Номер налогового документа для налоговых платежей
FKTaxDocDate	Текст	Дата документа для налоговых платежей

Тип атрибута	Тип значения	Описание
FKPaymentType	Текст	Тип платежа для налоговых платежей.
TOFK	Текст	Территориальное отделение Федерального казначейства
LSvUFK	Текст	Лицевой счет, открытый в ФК
ESIA_ID	Текст	Идентификатор пользователя ЕСИА
Shop_ID	Текст	Идентификатор точки продажи в системе Uniteller
SupplierFullName	Текст	Наименование плательщика
TreasureBranch	Текст	Наименование получателя
multyPurpose	Текст	Назначение платежа при оплате нескольких начислений одной операцией (мультиплатеж)
MultiNumber	Текст	Номер частного платежа в мультиплатеже
MultiTotal	Текст	Количество частных платежей в мультиплатеже
systemIdentifier	Текст	Универсальный идентификатор платежа

### 2.1.7.3 Возможные типы комиссий

Таблица 34. Возможные типы комиссий

Тип атрибута	Описание
2	Сумма комиссии «сверху»
3	Процент от суммы комиссии «сверху»
4	Сумма комиссии-вознаграждения 1 (от Поставщика начислений в пользу Расчётного банка)
7	Сумма комиссии-вознаграждения 2 (от Расчётного банка в пользу Поставщика начислений)
5	Сумма комиссии за ИТО
6	Сумма комиссии за услуги Предпроцессинга

### 2.1.7.4 Код причины отмены платежа

Таблица 35. Возможные значения для типа «Код причины отмены платежа»

Код	Название
USER_CANCEL	Отмена платежа на основании заявления пользователя
SUPPLIER_ERROR	Ошибка при приеме платежа поставщиком услуги
BANK_PAY_REQUEST_ERROR	Ошибка при передаче банку указания на перечисление денежных средств
BANK PAY ERROR	Ошибка при переводе банком денежных средств
BANK MONEY BACK	Возврат платежа банком получателя из-за неверных платёжных реквизитов
SUPPLIER MONEY BACK ERROR	Ошибка из-за возврата платежа поставщиком услуги
IPSH ERROR	Ошибка при обработке платежа в ИПШ
OVER ERROR	Прочие ошибки

### 2.1.7.5 Статус распоряжения на оплату в расчетном банке

Таблица 36. Возможные значения для типа «Статус распоряжения на оплату в расчетном банке»

Код	Название
NEW	Распоряжение передано в банк, но не выполнено

<b>Код</b>	<b>Название</b>
CREATE_ERROR	При создании распоряжения возникла ошибка из-за валидации переданных параметров
CANCELED	Распоряжение отменено, согласно запросу на отмены платежа
PROCESSING_ERROR	При выполнении распоряжения возникла ошибка из-за параметров распоряжения
DONE	Распоряжение выполнено
MONEY_BACK	По выполненному распоряжению были возвращены деньги
NOT_FOUND	Данные о платеже отсутствуют в полученной от банка сверке

## 2.2 Модуль «Центр обработки транзакций»

### 2.2.1 Протокол взаимодействия при оплате государственных услуг

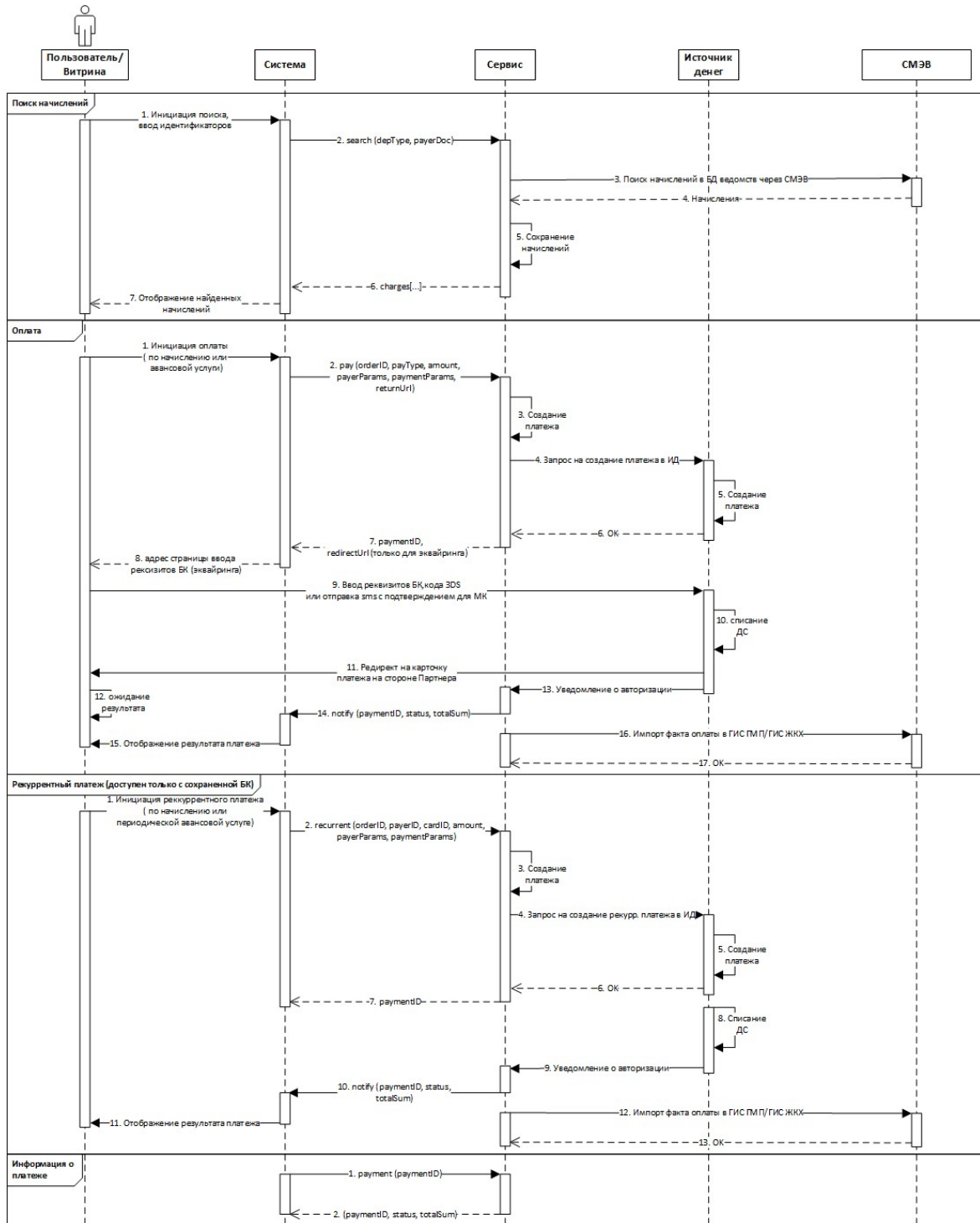


Рисунок 1. Протокол взаимодействия

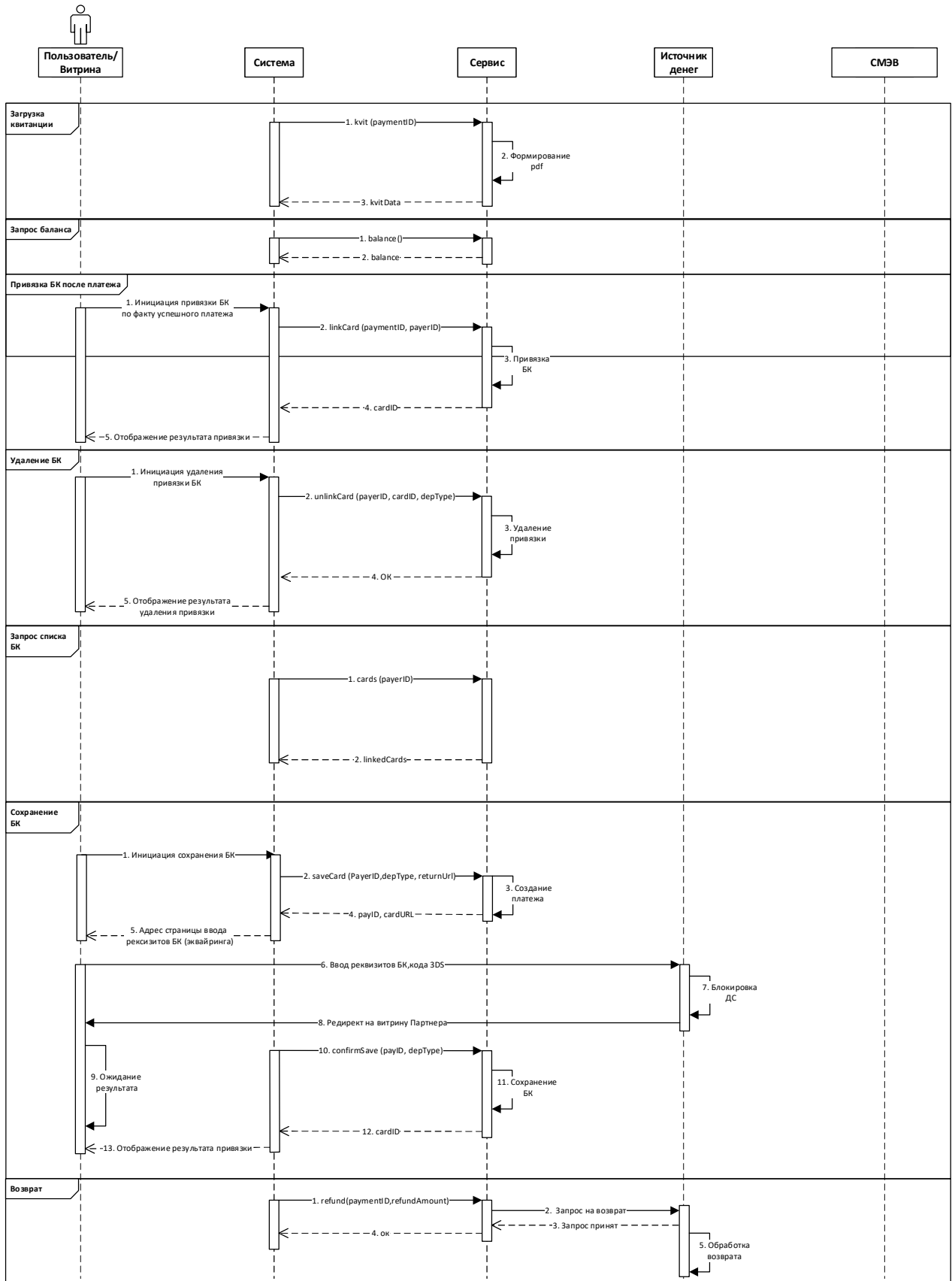


Рисунок 2. Протокол взаимодействия

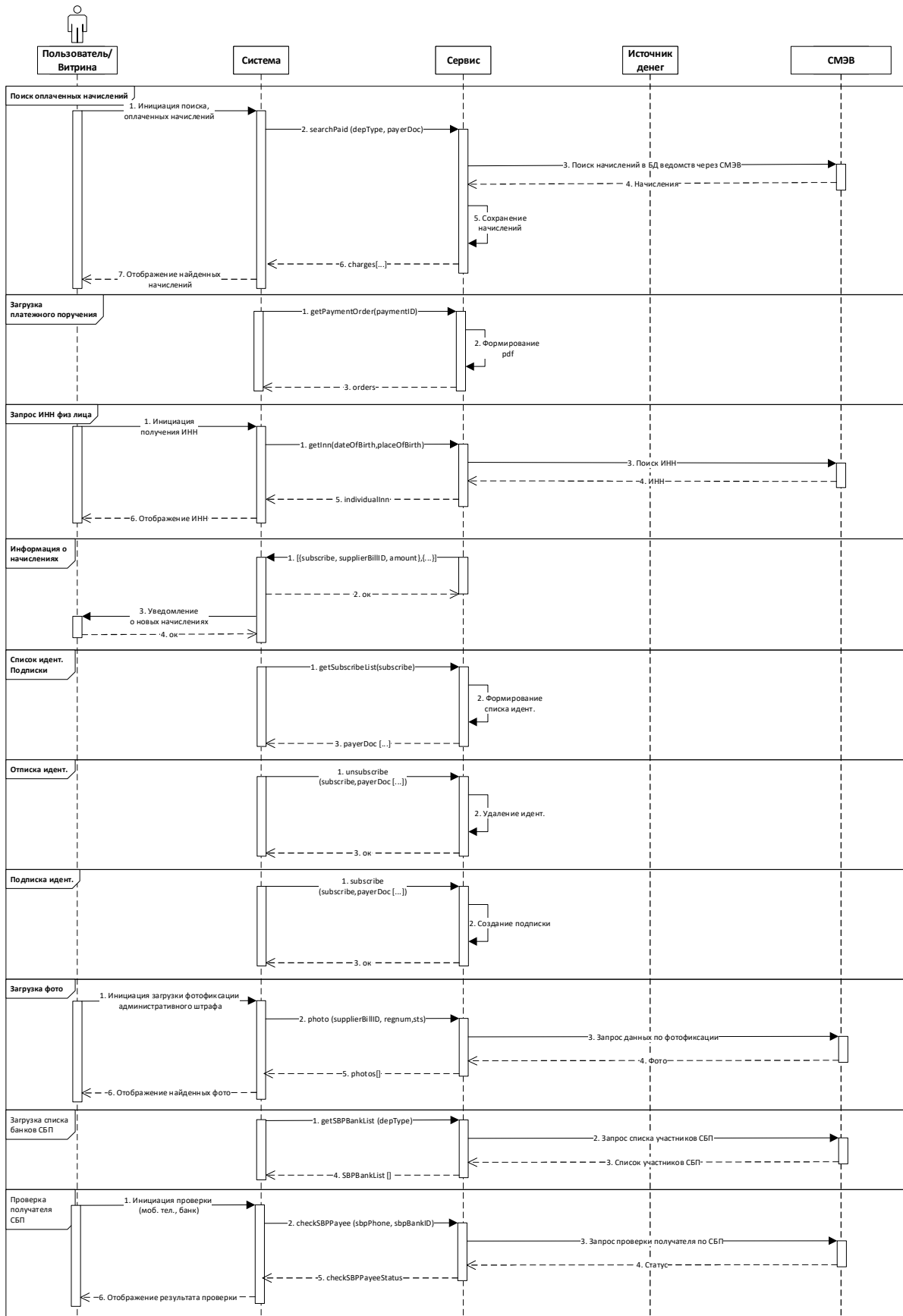


Рисунок 3. Протокол взаимодействия

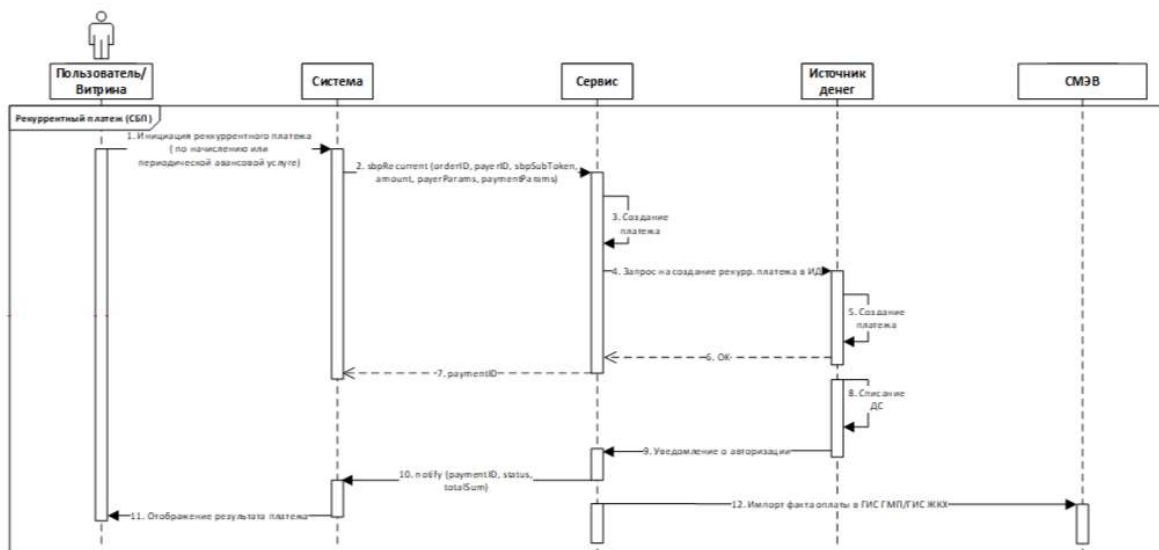


Рисунок 4. Протокол взаимодействия

## 2.2.2 Описание API

### 2.2.2.1 Метод search

Запрос Начислений по идентификаторам Пользователя или УИН.

### Схема взаимодействия

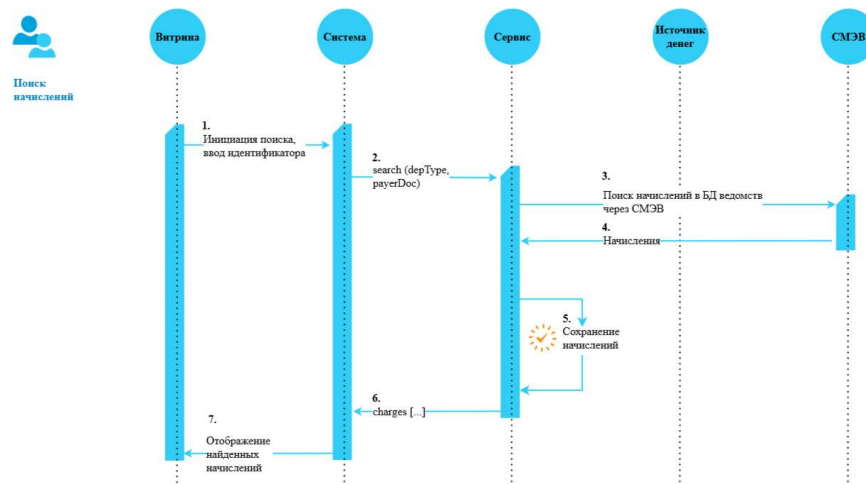


Рисунок 5. Схема взаимодействия при поиске начислений

Запрос Начислений по идентификаторам Пользователя или УИН. Поиск Начислений осуществляется асинхронно через среду СМЭВ в ГИС ГМП и информационных системах государственных ведомств. Для получения полных

данных по результатам поиска от всех источников необходимо повторять запрос до получения в ответе параметра status = complete.

Адрес запроса: POST http://demomultiserv.\*\*\*.ru/api/1.0/search

## Параметры

### Входные параметры:

Название	Значение	Об.	Описание
payerDoc	array	1	Идентификаторы Пользователя для осуществления поиска Начислений (см. подробнее тип <a href="#">payerDoc</a> ).
depType	string	0..1	Определяет тип ведомства поиска (см. тип <a href="#">depType</a> ). Если тип ведомства указан, Начисления будут отфильтрованы согласно этому ведомству. Если тип ведомства не указан, поиск вернет все найденные Начисления.
subscribe	string	0..1	Уникальный идентификатор Подписки Пользователя, в случае заполнения данного параметра создастся подписка на уведомления по новым/измененным/оплаченным/аннулированным начислениям, которые будут найдены по соответствующим Идентификаторам Пользователя, переданным в payerDoc. Если данный subscribe уже зарегистрирован в Сервисе, то новые payerDoc будут добавлены к уже существующей Подписке Пользователя.

### Выходные параметры:

Название	Значение	Описание
status	string	Статус выполнения поиска по переданным идентификаторам: <ul style="list-style-type: none"><li>part – частичный результат, ответ получен не от всех источников;</li><li>complete – полный результат, поиск завершен.</li></ul>
charges	array	Массив найденных начислений (см. подробнее тип <a href="#">charge</a> ).

## Ошибки:

Код ошибки	Описание
100	Возникла ошибка при выполнении поиска начисления.
101	Ошибка валидации входящего запроса.
122	Неверно сформирован JSON.

## Примеры

### Запрос для ГИБДД

#### Запрос:

```
curl -X POST --location "https://demopay.***.ru/api/multiserv/2.1/search" \  
-H "Content-Type: application/json; charset=utf-8" \  
-H "Authorization: Bearer testservice" \  
-d '{  
    "depType":"gibdd",  
    "payerDoc":[  
        {  
            "code":"22",  
            "value":[  
                "7712345678",  
                "7712345679"  
            ]  
        }  
    ],  
    "subscribe":"user12345"  
}'
```

#### Ответ:

```
{  
  "status":"complete",  
  "charges":[  
    {  
      "payerDoc":[  
        {  
          "code":"22",  
          "value":[  
            "7712345678"  
          ]  
        }  
      ]  
    }  
  ]  
}
```

```

    }
  ],
  "depType": "gibdd",
  "supplierBillID": "18810028150001578060",
  "chargeData": {
    "billDate": "2014-10-23T00:00:00Z",
    "amountToPay": 50000,
    "purpose": "ШТРАФПОАДМИНИСТРАТИВНОМУПРАВОНАРУШЕНИЮ",
    "kbk": "18811630020016000140",
    "oktmo": "45379000",
    "inn": "7707089101",
    "kpp": "770731005",
    "bik": "010349101",
    "accountNumber": "03100643000000011800",
    "payeeName": "ЦАФАПОДДГИБДДГУМВДРоссиипог.Москве",
    "divID": 12345,
    "amount": 50000,
    "payerName": "ИвановИванИванович",
    "treasureBranch": "УФКпоМосковскойобласти",
    "additionalData": {
      "discountSize": "50",
      "discountDate": "2016-10-20",
      "legalAct": "Часть 2 статьи 12.9 КоАП",
      "offenseName": "Превышение установленной скорости движения транспортного средства на величину более 20, но не более 40 км/ч",
      "offensePlace": "Г.САРАТОВ,УЛ.ОРДЖОНИКИДЗЕ,Д.24Б(ИЗЦЕНТРА)",
      "offenseDate": "2020-01-31 13:23:00",
      "offenseCoordinates": "43.920633,42.721550"
    }
  },
  "paymentData": {
    "sum": "10000",
    "date": "2016-10-23T18:00:00Z"
  },
  "ttl": "2017-02-25T21:00:00Z"
}
]
}

```

## Запрос для ГИБДД ЮЛ и ИП

### Запрос:

```

curl -i -X POST -H "Content-Type: application/json; charset=utf-8" -H
"Authorization: Bearer testservice" -d '{"depType": "gibdd", "payerDoc":
[{"code": "200", "value": ["7717751677771801001", "1650322494165001001"]},

```

```
{"code": "400","value": ["0000000771234567901"]}], "subscribe": "user12345"}'  
https://demopay.***.ru/api/multiserv/2.1/search
```

ОТВЕТ:

```
{  
  "status": "complete",  
  "charges": [  
    {  
      "payerDoc": [ {"code": "200", "value": ["7717751677771801001"]} ],  
      "depType": "gibdd",  
      "supplierBillID": "18810550220130045532",  
      "chargeData":  
        {  
          "billDate": "2021-10-23T00:00:00Z",  
          "amountToPay": 50000,  
          "purpose": "ШТРАФ ПО АДМИНИСТРАТИВНОМУ ПРАВОНАРУШЕНИЮ",  
          "kbc": "18811630020016000140",  
          "oktmo": "45379000",  
          "inn": "7707089101",  
          "kpp": "770731005",  
          "bik": "010349101",  
          "accountNumber": "03100643000000011800",  
          "payeeName": "ЦАФАП ОДД ГИБДД ГУ МВД России по г. Москве",  
            "divID": 12345,  
          "amount": 50000,  
          "payerName": "ООО Ромашка",  
          "payerIdentifier": "2007717751677771801001",  
          "additionalPayerIdentifier": "124000000009905896021"  
          "treasureBranch": "УФК по Московской области",  
          "additionalData":  
            {  
              "discountSize": "50",  
              "discountDate": "2016-10-20",  
              "legalAct": "Часть 2 статьи 12.9 КоАП",  
              "offenseName": "Превышение установленной скорости  
движения транспортного средства на величину более 20, но не более 40 км/ч",  
              "offensePlace": "Г. САРАТОВ, УЛ. ОРДЖОНИКИДЗЕ, Д.24Б  
(ИЗ ЦЕНТРА)",  
              "offenseDate": "2020-01-31 13:23:00",  
              "offenseCoordinates": "43.920633,42.721550"  
            },  
          },  
          "paymentData":  
            {  
              "sum": "10000",
```

```
"date": "2021-10-23T18:00:00Z"
  },
  "ttl": "2022-01-25T21:00:00Z"
}
]
}
```

## Запрос для ЖКХ

### Запрос:

```
curl -i -X POST -H "Content-Type: application/json; charset=utf-8" -H
"Authorization: Bearer testservice" -d '{"depType": "jkh", "payerDoc": [{"code":
"91", "value": ["11BA456331"]}]}
https://demopay.***.ru/api/multiserv/2.1/search
```

### Ответ

```
{
  "status": "complete",
  "charges": [
    {
      "payerDoc": [{"code": "91", "value": ["11BA456331"]}],
      "depType": "jkh",
      "paymentDocumentID": "11BA456331-03-7223",
      "chargeData": {
        "billDate": "2014-10-23T00:00:00Z",
        "amountToPay": 50000,
        "purpose": "Оплата ЛС //№604070002054491308//ИПД
//№11BA456331-03-7223// ЖКУ",
        "accountNumberJkh": "123450002053391444",
        "payPeriod": {
          "year": "2017",
          "month": "12"
        },
        "addressUserJkh": {
          "Region": "Иркутская обл.",
          "City": "Иркутская",
          "Street": "Терешковой",
          "HouseNum": "55",
          "FIASHouseGuid": "fd9c6c36-cc7b-4294-a20b-e22e4c8b3f4b",
          "Apartment": "55/a",
          "AddressString": "Иркутская обл., г. Иркутск, ул. Терешковой, д.
55, кв. 55/a"
        }
      }
    }
  ],
}
```

```

        "inn": "7707089101",
        "kpp": "770731005",
        "bik": "044583001",
        "bankName": "БАНК ВТБ (ПАО)",
        "accountNumber": "40101810800000010041",
        "correspondentBankAccount": "11112233300000111199",
        "payeeName": "ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ
        \УПРАВЛЯЮЩАЯ КОМПАНИЯ АНГАРА\",
        "mailingAddress": "angaratest@mail.ru"
    },

    "ttl": "2017-02-25T21:00:00Z"
}
]
}

```

### **Запрос поиска начисления Росреестра (юридическое лицо)**

#### Запрос:

```

curl -i -X POST -H "Content-Type: application/json; charset=utf-8" -H
"Authorization: Bearer testservice" -d '{"depType": "rosreestr", "payerDoc":
[{"code": "99", "value": ["32130080921833620712"]}]}
https://demopay.***.ru/api/multiserv/2.1/search

```

#### Ответ:

```

{
    "status": "complete",
    "charges": [
        {
            "unifiedPayerIdentifier": "26449013711644901001",
            "depType": "rosreestr",
            "supplierBillID": "18810028150001578060",
            "chargeData": {
                "billDate": "2014-10-23T00:00:00Z",
                "validUntil": "2014-10-31T00:00:00Z",
                "amountToPay": 50000,
                "purpose": "Плата за предоставление сведений, содержащихся в
                ЕГРН",
                "kbk": "18811630020016000140",
                "oktmo": "45379000",
                "inn": "7707089101",
                "kpp": "770731005",
                "bik": "044583001",
            }
        }
    ]
}

```

```

        "accountNumber ":"40101810800000010041",
        "payeeName":"УФК по Челябинской области (филиал ФГБУ «ФКП
Росреестра» по Челябинской области, л/с 20696У50940)",
        "divID":12345,
        "amount":50000,
        "treasureBranch":"УФК по Челябинской области"
    "paymentData":
    {
        "sum":"50000",
        "date":"2016-10-23T18:00:00Z"
    },
    "ttl":"2017-02-25T21:00:00Z"
}
]
}

```

### **Запрос поиска начисления ФССП (по ФИО)**

#### Запрос:

```

curl -i -X POST -H "Content-Type: application/json; charset=utf-8" -H
"Authorization: Bearer testservice" -d '{"depType": "fssp", "payerDoc":
[{"code": "931","value": ["ИВАНОВ РОМАН ИВАНОВИЧ 19710318"],"regionCode":78}]}'
https://demopay.***.ru/api/multiserv/2.1/search

```

#### Ответ:

```

{
    "status":"complete",
    "charges":[
        {
            "payerDoc": [{"code": "931","value": ["ИВАНОВ РОМАН ИВАНОВИЧ
19710318"],"regionCode": 78}],
            "depType":"fssp",
            "supplierBillID ":"32250061180619679006",
            "chargeData":
            {
                "billDate":"2014-10-23T00:00:00Z",
                "amountToPay":50000,
                "purpose":"Штраф ГИБДД",
                "kbc":"18811630020016000140",
                "oktmo":"45379000",
                "inn":"7707089101",
                "kpp":"770731005",
                "bik":"044583001",
                "accountNumber ":"40101810800000010041",

```

```

        "payeeName": "УФК по Московской области (МОСП по ВАШ №1 УФССП
России по Московской области, л/с 05481D10630)",
        "divID": 12345,
        "amount": 50000,
        "treasureBranch": "УФК по Московской области",
    },
    "paymentData":
    {
        "sum": "50000",
        "date": "2016-10-23T18:00:00Z"
    },
    "ttl": "2017-02-25T21:00:00Z"
}
]
}

```

**Запрос на получение суммы задолженностей за проезды по платным дорогам с системой взимания платы "Свободный поток"**

Запрос:

```

curl -i -X POST -H "Content-Type: application/json; charset=utf-8" -H
"Authorization: Bearer testservice" -d '{"depType": "freeFlow", "payerDoc":
[{"code": "94", "value": ["11BA456331"], "country": "RUS"}]}'
https://demopay.***.ru/api/multiserv/2.1/search

```

Ответ:

```

{
    "status": "complete",
    "charges": [
        {
            "payerDoc": [{"code": "94", "value": ["11BA456331"], "country": "RUS"}],
            "depType": " freeFlow",
            "chargeData":
            {
                "amount": 7000,
                "plazaTransactions": [395064880]
            },
            "ttl": "2017-02-25T21:00:00Z"
        }
    ]
}
]
}

```

**Запрос для проезда по платным дорогам из ГИС ГМП**

### Запрос:

```
curl -i -X POST -H "Content-Type: application/json; charset=utf-8" -H
"Authorization: Bearer testservice" -d '{"depType": "paidRoads", "payerDoc":
[{"code": "15", "value": ["RUSA001AA77"]}], "subscribe": "user12345" }'
https://demopay.***.ru/api/multiserv/2.1/search
```

### Ответ:

```
{
  "status": "complete",
  "charges": [
    {
      "payerDoc": [ {"code": "15", "value": ["RUSA001AA77"]} ],
      "depType": "paidRoads",
      "supplierBillID": "0412479001000013586054242",
      "chargeData":
      {
        "billDate": "2024-09-03T09:33:56.000+03:00",
        "amountToPay": 18700,
        "purpose": "Оплата проезда по маршруту А-113 ЦКАД (М-12 - Егорьевское
шоссе) от 03.09.2024, ГРН У705ВТ44RUS НДС не облагается",
        "kbk": "0",
        "oktmo": "0",
        "inn": "7710965662",
        "kpp": "770701001",
        "ogrn": "770701001",
        "bik": "044525225",
        "bankName": "ПАО Сбербанк",
        "accountNumber": "40702810940000425661",
        "correspondentBankAccount": "30101810400000000225",
        "payeeName": "ООО АВТОДОР - ПЛАТНЫЕ ДОРОГИ",
        "amount": 18700,
        "additionalData":
        {
          "legalAct": "Договор-оферта от 05.01.2022 №АД/35-
01",
          "offensePlace": "А-113 ЦКАД, М-12 - Егорьевское
шоссе",
          "offenseDate": "2024-09-03T09:33:56+03:00",
          "passNumber": "1358605424"
        },
      },
    },
  ],
  "ttl": "2024-09-05T21:00:00Z"
}
```

### 2.2.2.2 Метод pay

Инициация оплаты по Начислению или Авансового платежа.

Адрес запроса: POST [http://demomultiserv.\\*\\*\\*.ru/api/1.0/pay](http://demomultiserv.***.ru/api/1.0/pay)

#### Схема взаимодействия

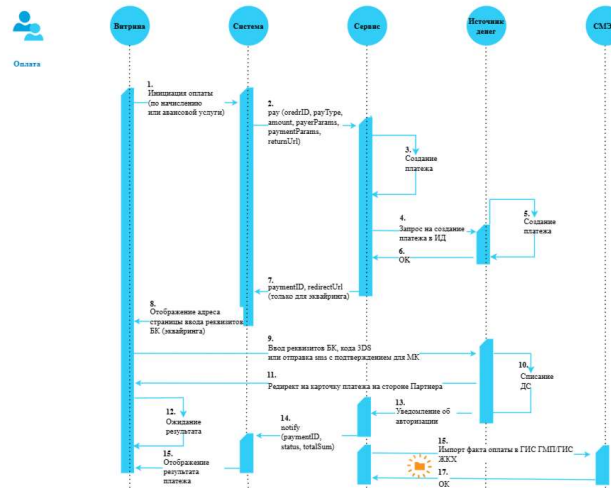


Рисунок 6. Схема взаимодействия при оплате

#### Параметры

Входные параметры:

Название	Значение	Об.	Описание
orderID	string	1	Уникальный номер заказа на стороне Партнера.
depType	string	0..1	Наименование ведомства Получателя. Необязательно при указании в запросе УИН (supplierBillID !=0 в paymentParams) и для выплаты на СБП.
payType	string	0..1	Способ оплаты, возможные значения: <ul style="list-style-type: none"> <li>• phone – со счета мобильного телефона;</li> <li>• card – банковской картой;</li> <li>• deposit – со счета обеспечения Партнера;</li> <li>• pscb – со счета обеспечения в АО “ПСКБ”;</li> </ul>

			<ul style="list-style-type: none"> <li>• yandexpay - через сервис Yandex Pay;</li> <li>• sbp – через сервис быстрых платежей ЦБ РФ;</li> <li>• b2csbpay – выплата на СБП со счета обеспечения Партнера;</li> <li>• sbp2sbp – выплата с СБП на СБП для ФЛ;</li> </ul> <p>Для card, sbp, googlepay, applepay и yandexpay доступна оплата одним платежом нескольких начислений.</p> <p>Для card, sbp, googlepay, applepay и yandexpay доступна оплата одним платежом нескольких начислений.</p> <p>Можно не заполнять при оплате с использованием платежной страницы Сервиса, на которой доступны: банковская карта, Yandex Pay, СБП.</p>
defPayType	string	0..1	<p>Способ оплаты по умолчанию на платежной странице Сервиса, доступные значения:</p> <ul style="list-style-type: none"> <li>• card;</li> <li>• yandexpay;</li> <li>• sbp.</li> </ul>
linkCardCheckBoxState	boolean	0..1	<ul style="list-style-type: none"> <li>• true - устанавливает чекбокс "Сохранить карту" на UI в состоянии 'активен';</li> <li>• false - устанавливает чекбокс "Сохранить карту" на UI в состоянии 'не активен';</li> <li>• Не установлено - устанавливает чекбокс "Сохранить карту" на UI в состоянии 'активен'.</li> </ul>
allowPayTypes	array	0..1	<p>В массив передаются значения доступных для отображения на UI способов оплаты:</p> <ul style="list-style-type: none"> <li>• card;</li> <li>• yandexpay;</li> <li>• sbp.</li> </ul> <p>На витрине не отображаются способы оплаты, не перечисленные в массиве.</p> <p><b>Важно!</b></p> <p>Если указан <i>defPayType</i> и в нём указан способ оплаты, указанный в <i>allowPayTypes</i> - он выбирается по умолчанию.</p>

			<p>Если в <i>defPayType</i> указан способ оплаты, не указанный в <i>allowPayType</i> - он игнорируется.</p> <p>Если в массиве перечислен ненастроенный партнёру способ оплаты - он игнорируется.</p> <p>Если в массиве есть только способы оплаты, которые не настроены на товаре - они игнорируются, отображаются только все доступные.</p>
kvit	boolean	1	<p>Заказ квитанции, возможные значения:</p> <ul style="list-style-type: none"> <li>• true – с квитанцией;</li> <li>• false – без квитанции.</li> </ul>
amount	int	0..1	Сумма платежа без учета комиссий, обязательна для Авансовых платежей и при частичной оплате Начисления, указывается в копейках. Может превышать сумму начисления.
totalSum	int	0..1	Сумма платежа с учетом комиссий, которая будет списана с Пользователя, в копейках, используется при расчете суммы списания с Пользователя ИС Партнера, использование параметра возможно только после согласования с менеджером. В случае использования с мультиплатежами должна быть указана для каждой операции.
payerParams	object	1	Данные о Пользователе (см. подробнее тип payerParams).
paymentParams	object	1	Данные по оплачиваемому Начислению и Получателю платежа (смотри подробнее тип paymentParams).
payerID	string	0..1	<p>Длина 28 символов (латинских букв/цифр).</p> <p>Идентификатор Пользователя в информационной системе Партнера. Значение должно быть уникальным для каждого Пользователя. Параметр обязателен для заполнения при оплате с привязанной Банковской карты.</p> <p><b>Note! Параметр обязателен для подписки СБП.</b></p>
sbpSubscription	object	0..1	Привязка счета при оплате через СБП, для последующей рекуррентной оплаты без акцепта плательщика

			(метод sbpRecurrent). Используется совместно с payType = sbp. payerID должен быть заполнен.
enable	boolean	1	Активация привязки счета СБП. Возможные значения: "true" – привязать счет к payerID, "false" – не привязывать счет к payerID.
purpose	string	0..1	Назначение подписки, передаётся при оформлении подписки в СБП.
cardID	string	0	Идентификатор привязанной к payerID банковской карты.
cryptogram	string	0..1	<p>Зашифрованные данные банковской карты. При оплате с привязанной карты (заполнены payerID и cardID) в криптограмме заполняется только параметр cvv.</p> <p>Если используется phone вместо payerID, то передача параметра phone обязательна.</p> <p>См. описание формирования криптограммы.</p>
mpiExtInfo	string	0..1	<p>Информация об устройстве Пользователя, обязательно при оплате с криптограммой, applepay, googlepay, yandexpay. Передается в виде строки, содержащей JSON со следующими полями:</p> <ul style="list-style-type: none"> <li>• browserAcceptHeader - Содержимое HTTP-заголовка Accept. Максимальная длина – 2048 символов;</li> <li>• browserColorDepth - Значение, представляющее битовую глубину цветовой палитры;</li> <li>• browserIp - IP-адрес браузера;</li> <li>• browserLanguage - Язык браузера, указанный в IETF BCP47. Максимальное значение – 8 символов;</li> <li>• browserScreenHeight - Общая высота (в пикселях) экрана, отображаемого держателю карты. Максимальное значение – 6 символов;</li> <li>• browserScreenWidth - Общая ширина (в пикселях) экрана, отображаемого держателю карты. Максимальное значение – 6 символов;</li> <li>• browserTz - Разница во времени между временем по UTC и местным временем браузера пользователя. Максимальное значение – 5 символов;</li> </ul>

			<ul style="list-style-type: none"> <li>• browserUserAgent - Содержимое HTTP-заголовка User-Agent. Максимальная длина – 2048 символов;</li> <li>• deviceChannel - Тип устройства, с которого инициирована транзакция. Возможные значения: <ul style="list-style-type: none"> <li>• 01 - мобильное приложение Партнера;</li> <li>• 02 - браузер пользователя;</li> <li>• 03 - 3DS Requestor;</li> </ul> </li> <li>• browserJavaEnabled - Признак возможности выполнения JavaScript в браузере держателя карты. Возможные значения: <ul style="list-style-type: none"> <li>• true;</li> <li>• false;</li> </ul> </li> <li>• windowWidth - Ширина окна браузера (в пикселях), в котором отображаются страницы сайта Партнера;</li> <li>• windowHeight - Высота окна браузера (в пикселях), в котором отображаются страницы сайта Партнера.</li> </ul>
appleToken	string	0..1	<p>Платёжный токен от Apple Pay. Необходимо преобразовать JSON строку, полученную от Apple Pay, в строку Base64. Для использования оплаты через Apple Pay Партнер должен предоставить сертификат ключа в формате elliptic curve и merchantId.</p> <p><i>Пример:</i></p> <pre>{   "header": {     "publicKeyHash": "publicKeyHash",     "ephemeralPublicKey": "ephemeralPublicKey",     "transactionId": "transactionId"   },   "signature": "signature",   "data": "data",   "version": "EC_v1" }</pre>
gpToken	string	0..1	<p>Платёжный токен от Google Pay. Необходимо преобразовать JSON строку в строку Base64.</p>
ypToken	string	0..1	<p>Платёжный токен от Yandex Pay, строка должна быть кодирована в формате base64.</p>
returnUrl	string	0..1	<p>Адрес Витрины для возврата Пользователя на сайт партнера («Назад в магазин»). Будет использован для переадресации по фактам успешного или неуспешного платежа, если не заполнены returnUrlSuccess или returnUrlFail.</p>

returnSuccessUrl	string	0..1	Адрес Витрины для возврата Пользователя после успешного платежа.
returnFailUrl	string	0..1	Адрес Витрины для возврата Пользователя после неуспешного платежа.
sign	string	1	Подпись запроса. Формируется в зависимости от платежа. Подробнее см. в разделе Подпись (sign). Секретное слово индивидуально для каждого Партнера.

### Выходные параметры:

Название	Значение	Об.	Описание
paymentID	int	1	Идентификатор Платежа в ПЦ.
redirectUrl	string	0..1	Адрес для редиректа Пользователя на страницу ввода платежных реквизитов. (необязательный).
3ds	object	0..1	При наличии в ответе данного параметра Пользователя необходимо направить на адрес, указанный в acsUrl и передать следующие параметры в зависимости от типа 3ds-аутентификации: <ul style="list-style-type: none"> <li>• TermUrl;</li> <li>• PaReq;</li> <li>• MD;</li> </ul> или <ul style="list-style-type: none"> <li>• cseq;</li> </ul> или без параметров в случае отсутствия PaReq или cseq. Названия параметров чувствительны к регистру.
acsUrl	string	0..1	Адрес для перенаправления Пользователя на сервис 3ds-аутентификации эмитента.

termUrl	string	0..1	Адрес возврата Пользователя после 3ds-аутентификации эмитента.
paReq	string	0..1	Закодированные данные для 3ds-аутентификации.
creq	string	0..1	Закодированные данные для 3ds-аутентификации.
md	string	0..1	Закодированные данные для 3ds-аутентификации.
sbpCode	string	0..1	Текстовое содержимое QR-кода СБП, которое партнер может использовать для самостоятельной генерации изображения QR-кода.
sbpCodeImageUrl	string	0..1	Ссылка на файл с изображением QR-кода СБП.
payloadSbpToSbp	string	0..1	Платежная ссылка для переводов СБП на СБП.
QRImgSbpToSbp	string	0..1	Байтовое представление картинка с QR-кодом в виде строки base64 для переводов СБП на СБП.

## Примеры

### Запрос на оплату начисления:

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer testservice" -d '{
  "orderId": "12345",
  "depType": "gibdd",
  "payType": "card",
  "kvit": true,
  "payerParams": {
    "fio": "Иванов Иван Иванович"
```

```
    },  
    "paymentParams": {  
      "supplierBillID": "18810177170712879661"  
    },  
    "returnUrl": "https://testpartner.ru/e0462fc0872e2/",  
    "sign": "d94c3157500e45ffa87261c05c64d258"  
  }'  
'
```

https://demopay.\*\*\*.ru/api/multiserv/2.0/pay

Пример формирования параметра sign:

1. Сформировать строку из значений параметров согласно описанию поля sign.

- Если параметр имеет вложения, то добавляются значения всех вложений в порядке, указанном в документации;
- Значение параметра kvit должно быть приведено к int (true – 1, false – 0).

Сформированная строка для примера запроса на оплату начисления:

card118810177170712879661

2. Добавить в конец строки из п. 1 секретное слово Партнера. На тестовой зоне:

SA9QXHKV: card118810177170712879661SA9QXHKV

3. От полученной в п. 2 строки получить md5 хеш. В итоге sign:

d94c3157500e45ffa87261c05c64d258.

Запрос на оплату начисления через сервис Yandex Pay:

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization:  
Bearer testservice" -d '  
{  
  "orderID": "12345",  
  "depType": "gibdd",  
  "payType": "yandexpay",  
  "kvit": true,  
  "payerParams": {  
    "fio": "Иванов Иван Иванович"  
  },  
  "paymentParams": {
```



```
ENJRjQ2Y3VkT1FKT2dQdlE3Q1N2cWtPYWVISIFYUEYvUlerSVZtNkhGQ29IT0c2
RGtLVfmg4b050MjNob0JaaEdMUmpXd094az1cIn0iLCJwcm90b2NvbFZlcnNpb24iOi
JFQ3YyIiwic2lnbmF0dXJlIjoiTUVZQ0lRQ2hRYkJBaFN2bXA0U0JMTzRtLzc2aFA5
MVBjTS9Qc0dPSG8zZnQ2TzczRHdJaEFQOUV0VIRtdENGRHkyYVpNdE9ZK3Y1
RXdiZEd2aG92dmFXRnIxQjkrS09mIiwiaW50ZXJtZWVpYXRlU2lnbmluZ0tleSI6eyJ
zaWduZWRLZXkiOiJ7XCJrZXIwYX1ZVwiOlwiTUZrd0V3WUhlb1pJemowQ0FR
WUilb1pJemowREFRY0RRZ0FFb1Zjb01scW9ZUG8rYyt5QWU0UkU0eHh6ZktDW
khjQlF0Vkp0QlZYakJraTRYgeENLeXNrcEluMmYzTGIBa0JVYXhnYXJmalR6bTAz
c1JjTjdOd3NFWIE9PVwiLFwia2V5RXhwaXJhdGlvblwiOlwiMTc2NTcwNDA0NTA
wMFwifSIsInNpZ25hdHVyZXMiOlsiTUVVQ0lRQyYtcVZWZwXpMFnjWTVNVVm85
Y0VoWW5UMGl3eTlmM0YxK2QxdXFDR0kvdGdJZ0Q0dk9oU3pIVTIzUTBZcmhz
TkNOSnZtMXVORG9mMmRPTlpYMnh6RFo2RTQ9Il19fQ==",
"returnUrl": "https://testpartner.ru/e0462fc0872e2/",
"sign": "9192ab4b5a622b94fde0d2dbc52bfde3"
}'
https://demopay.***.ru/api/multiserv/2.0/pay
```

### Запрос на оплату начислений (мультиплатеж):

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer
testservice" -d '
{
"orderID": "12345",
"depType": "gibdd",
"payType": "card",
"kvit": true,
"payerParams": {
"phone": "9111234567",
"fiо": "Иванов Иван Иванович"
},
"paymentParams": {
"supplierBillID": ["18810177170712879661","18810177170712879111"]
},
"returnUrl": "https://testpartner.ru/e0462fc0872e2/",
```

```
"sign": "d94c3157500e45ffa87261c05c64d258"  
'  
https://demopay.***.ru/api/multiserv/2.0/pay
```

#### Запрос на оплату авансового платежа:

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer  
testservice" -d '  
{  
  "orderID": "12345",  
  "depType": "gibdd",  
  "payType": "card",  
  "kvit": true,  
  "payerParams": {  
    "phone": "9111234567",  
    "fio": "Иванов Иван Иванович"  
  },  
  "paymentParams": {  
    "supplierBillID": ["18810177170712879661","18810177170712879111"]  
  },  
  "returnUrl": "https://testpartner.ru/e0462fc0872e2/",  
  "sign": "d94c3157500e45ffa87261c05c64d258"  
}'  
https://demopay.***.ru/api/multiserv/2.0/pay  
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer  
testservice" -d '  
{  
  "orderID": "1234567891",  
  "depType": "gibdd",  
  "payType": "card",  
  "kvit": false,  
  "amount": 10000,  
  "payerParams": {  
    "phone": "9111234574",
```

```
"fio":"Иванов Иван Иванович",
"address":"Школьная 25",
"email":"aivanov@tprs.ru",
"payerDoc":[{
  "code":"22",
  "value":["1234567890"]
}],
"paymentParams":{
  "purpose":"УИН18810067170000804114///ШТРАФ ПО АДМИНИСТРАТИВНОМУ
ПРАВОНАРУШЕНИЮ, № 18810067170000804114 от 17.07.2017, тест тест тест",
  "kbk":"18210807010011000110",
  "oktmo":"45367000",
  "inn":"7733506810",
  "kpp":"773301001",
  "bik":"048327001",
  "accountNumber":"40101810800000010041",
  "payeeName":"УФК по г. Москве (МИФНС России № 46 по г. Москве)"
},
"returnUrl":"https://testpartner.ru/e0462fc0872e9/",
"sign":"f1e4803a872c6a18299520c617063ec3"
}'
https://demopay.***.ru/api/multiserv/2.0/pay
```

#### Запрос на оплату пошлины ФНС:

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer
testservice" -d '
{
  "orderId": "6280536",
  "depType": "fns",
  "payType": "deposit",
  "kvit": false,
  "amount": 80000,
```

```
"payerParams": {
  "fio": "Иванов Иван Иванович",
  "address": "г. Москва, ул. Ботаническая, Дом 19, Корпус 1, Квартира 172",
  "payerDoc": [
    {
      "code": "21",
      "value": [
        "352500181687"
      ]
    }
  ],
},
"paymentParams": {
  "purpose": "Иванов Иван Иванович, 352500181687, Государственная пошлина за
регистрацию ФЛ в качестве ИП",
  "kbk": "18210807010011000110",
  "oktmo": "45367000",
  "inn": "7733506810",
  "kpp": "773301001",
  "bik": "048327001",
  "accountNumber": "40101810800000010041",
  "payeeName": "УФК по г.Москве (МИ ФНС России No 46 по г.Москве)"
},
"sign": "d057bc4ef9bf925d3bfa3673f96b0f91"
}'
https://demopay.***.ru/api/multiserv/2.0/pay
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer
testservice" -d '
{
  "orderID": "6280536",
  "depType": "fns",
  "payType": "deposit",
  "kvit": false,
  "amount": 80000,
```

```
"payerParams": {
  "fio": "Иванов Иван Иванович",
  "address": "г. Москва, ул. Ботаническая, Дом 19, Корпус 1, Квартира 172",
  "payerDoc": [
    {
      "code": "21",
      "value": [
        "352500181687"
      ]
    }
  ],
},
"paymentParams": {
  "purpose": "Иванов Иван Иванович, 352500181687, Государственная пошлина за
регистрацию ФЛ в качестве ИП",
  "kbk": "18210807010011000110",
  "oktmo": "45367000",
  "inn": "7733506810",
  "kpp": "773301001",
  "bik": "048327001",
  "accountNumber": "40101810800000010041",
  "payeeName": "УФК по г.Москве (МИ ФНС России No 46 по г.Москве)"
},
"sign": "d057bc4ef9bf925d3bfa3673f96b0f91"
}'
https://demopay.***.ru/api/multiserv/2.0/pay
```

#### Запрос на оплату начисления ЖКХ по ИПД:

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer
testservice" -d '
{
"orderID": "12345",
```

```
"depType": "jkh",
"payType": "card",
"kvit": false,
"payerParams": {
  "fio": "Иванов Иван Иванович"
},

"paymentParams": {
  "paymentDocumentID": "11BA456331-03-7223"
},
"payerID": "user11111122233334456782345",
"returnUrl": "https://testpartner.ru/e0462fc0872e2/",
"sign": "ff85a75dcebc55cb5bbd640a2cdfc72e"
}'
```

[https://demopay.\\*\\*\\*.ru/api/multiserv/2.0/pay](https://demopay.***.ru/api/multiserv/2.0/pay)

Запрос на оплату начисления ЖКХ с ИПД и полными реквизитами:

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer testservice" -d '
```

```
{
"orderID": "12345",
"depType": "jkh",
"payType": "card",
"kvit": false,
"payerParams": {
  "fio": "Иванов Иван Иванович"
},
"paymentParams": {
  "amount": 10000,
  "purpose": "ИПД № 10АО223172-10-2051, ЖКУ, Иванов Иван Иванович",
```

```
"inn": "6167111598",
"kpp": "616701001",
"bik": "044525411",
"correspondentBankAccount": "30101810145250000411",
"accountNumber": "03100643000000017300",
"payeeName": "НКО ФОНД КАПИТАЛЬНОГО РЕМОНТА",
"accountNumberJkh": "604370002067170408",
"paymentDocumentID": "10АО223172-10-2051"
},
"payerID": "user11111122233334456782345",
"returnUrl": "https://testpartner.ru/e0462fc0872e2/",
"sign": "e3ec60a7e16ecc1fe18b9689695585ef"
}'

https://demopay.***.ru/api/multiserv/2.0/pay
```

Запрос на оплату начисления ЖКХ по ИПД (мультиплатеж):

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer testservice" -d '{
"orderID": "12345",
"depType": "jkh",
"payType": "card",
"kvit": false,
"payerParams": {
  "fio": "Иванов Иван Иванович"
},
"paymentParams": {
  "paymentDocumentID":["11BA456331-03-7223","11BA456441-03-7223"]
},
"payerID": "user11111122233334456782345",
"returnUrl": "https://testpartner.ru/e0462fc0872e2/",
```

```
"sign": "ff85a75dcebc55cb5bbd640a2cdfc72e"  
}'
```

[https://demopay.\\*\\*\\*.ru/api/multiserv/2.0/pay](https://demopay.***.ru/api/multiserv/2.0/pay)

Запрос на оплату начисления ЖКХ с ИПД по полным реквизитам  
(мультиплатеж):

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer  
testservice" -d '  
{  
  "orderID": "12345",  
  "depType": "jkh",  
  "payType": "card",  
  "kvit": false,  
  "payerParams": {  
    "fio": "Иванов Иван Иванович"  
  },  
  "paymentParams": {  
    "packetAdvancePayments": [  
      {  
        "amount": 10000,  
        "purpose": "ИПД № 10АО223172-10-2051, ЖКУ, Иванов Иван Иванович",  
        "inn": "6167111598",  
        "kpp": "616701001",  
        "bik": "044525411",  
        "correspondentBankAccount": "30101810145250000411",  
        "accountNumber": "40703810200300000220",  
        "payeeName": "НКО ФОНД КАПИТАЛЬНОГО РЕМОНТА",  
        "accountNumberJkh": "604370002067170408",  
        "paymentDocumentID": "10АО223172-10-2051"  
      },  
      {  
        "amount": 15000,
```

```
"purpose": "ИПД № 50ЕТ610781-04-2031, ЖКУ, Иванов Иван Иванович",
"inn": "5410772955",
"kpp": "540701001",
"bik": "044525360",
"correspondentBankAccount": "30101810445250000360",
"accountNumber": "40702810512010580258",
"payeeName": "ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ
ЭКОЛОГИЯ-НОВОСИБИРСК",
"accountNumberJkh": "13202202194",
"paymentDocumentID":"50ЕТ610781-04-2031"
}
],
},
"payerID": "user11111122233334456782345",
"returnUrl": "https://testpartner.ru/e0462fc0872e2/",
"sign": "17876c0b03a896022fd9ed47bf85bc91"
}'

https://demopay.***.ru/api/multiserv/2.0/pay
```

Запрос на оплату авансового платежа ЖКХ с привязанной банковской карты:

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer
testservice" -d '
{
"orderId": "12345",
"depType": "jkh",
"payType": "card",
"kvit": false,
"amount": 324512,
"payerParams": {
"fiо": "Иванов Иван Иванович",
"email": "ivanov@mail.ru",
"accountNumber": "1234567890"
```

```
    },
    "paymentParams": {
      "purpose": "Оплата услуг ЖКХ, Иванов Иван Иванович, ул. Мира, д. 2, кв 3, л.с.
1234567890, за 12.2017",
      "payPeriod": {
        "year": "2017",
        "month": "12"
      }
    },
    "payerID": "user11111122233334456782345",
    "returnUrl": "https://testpartner.ru/e0462fc0872e2/",
    "sign": "4c65b76f023c8fac61bdef383ab8fc6c"
  }'
https://demopay.***.ru/api/multiserv/2.0/pay
```

Запрос на оплату начисления с криптограммой:

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization:
Bearer testservice" -d '
{
  "orderID": "12345",
  "depType": "gibdd",
  "payType": "card",
  "kvit": true,
  "payerParams": {
    "phone": "9111234567",
    "fio": "Иванов Иван Иванович"
  },
  "paymentParams": {
    "supplierBillID": "18810177170712879661"
  },
  "cryptogram": "*****",
  "mpiExtInfo": "{
```

```

"browserAcceptHeader": "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*\n/*;q=0.8",
"browserIp": "127.0.0.1",
"browserLanguage": "ru",
"browserScreenHeight": 1080,
"browserScreenWidth": 1920,
"browserTz": "-180",
"browserUserAgent": "Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.019 Safari/537.36",
"deviceChannel": "02",
"browserJavaEnabled": "false",
"windowWidth": 1920,
>windowHeight": 1080,
"browserColorDepth": 24
}",
"returnUrl": "https://testpartner.ru/e0462fc0872e2/",
"sign": "d94c3157500e45ffa87261c05c64d258"
}'
https://demopay.***.ru/api/multiserv/2.0/pay

```

Для тестовых целей, на странице ввода БК можно указать код SVC (для получения определенного ответа от эмулятора). Доступны следующие коды:

- 000 - задержка при оплате 5-10 минут;
- 111 - ошибка на привязку БК;
- 666 - ошибка на первый processPayment;
- 668 - ошибка на второй processPayment;
- 777 - платёж без 3DS;
- 555 - эмулятор на статус возвращает "in progress" и уходит в ожидание на 5 мин;
- 444 - Передача пустого параметра MD с формы 3DS (реализовано для оплаты с БК МИР);
- \*\*\* - все остальные (любые) обычная оплата с криптограммой и 3DS.

Запрос на пополнение парковочного счета ГКУ АМПП:

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer
testservice" -d '
{
"orderId": "12345",
"depType": "rnip",
"payType": "card",
"kvit": false,
"amount":10000,
"payerParams": {
  "fio": "Иванов Иван Иванович",
  "payerDoc":[{
    "code":"22",
    "value":["1234567890"]
  }]
},
"paymentParams": {
  "purpose": "Плата за пользование парковочными местами. НДС не облагается., №
79111234567",
  "additionalData":
    {
      "SRV_CODE":"AAAA363C060000123456",
      "NUM_TELEFON":"79111234567"
    },
},
"returnUrl": "https://testpartner.ru/e0462fc0872e2/",
"sign": "ca13360d4a828cdf023ab19afd1ea009"
}'

https://demopay.***.ru/api/multiserv/2.0/pay
```

Запрос на оплату ГИБДД (мультиплатеж):

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer
testservice" -d '
```

```
{
"orderID": "12345",
"depType": "gibdd",
"payType": "card",
"kvit": true,
"payerParams": {
  "fio": "Иванов Иван Иванович"
},
"paymentParams": {
  "supplierBillID": ["18810177170712879661","18810110180623814000"]
},
"returnUrl": "https://testpartner.ru/e0462fc0872e2/",
"sign": " 9036fa1c804fe01e42f8d1cedd352a1e"
}'
https://demopay.***.ru/api/multiserv/2.0/pay
```

#### Запрос на оплату услуги Росреестра по реквизитам:

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer testservice" -d '{
"orderID": "6280536",
"depType": "rosreestr",
"payType": "card",
"kvit": false,
"amount": 40000,
"payerParams": {
  "fio": "Иванов Иван Иванович",
  "address": "г. Москва, ул. Ботаническая, Дом 19, Корпус 1, Квартира 172",
  "payerDoc": [
    {
      "code": "01",
      "value": [ "1234567890" ]
    }
  ]
}
```

```
]
},
"paymentParams": {
  "purpose": "Оплата Видеолекции №1, Иванов Иван Иванович, паспорт РФ
1234567890",
  "kbk": "0000000000000000130",
  "oktmo": "45378000",
  "inn": "770801001",
  "kpp": "7705401340",
  "bik": "004525988",
  "correspondentBankAccount": "40102810545370000003",
  "accountNumber": "03100643000000017300",
  "payeeName": "УФК по г.Москве (ФГБУ «ФКП Росреестра»)"}
},
"sign": "a72b10c451b6c50a6df0dbad2c6a5e04"
}'
https://demopay.***.ru/api/multiserv/2.0/pay
```

Запрос на пакетную оплату услуг Росреестра авансом (по реквизитам):

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer
testservice" -d '
{
  "orderID": "6280536",
  "depType": "rosreestr",
  "payType": "card",
  "kvit": false,
  "amount": 40000,
  "payerParams": {
    "fio": "Иванов Иван Иванович",
    "payerDoc": [{
      "code": "01",
      "value": ["1234567890"]}]}
  },
}
```



```
}  
https://demopay.***.ru/api/multiserv/2.0/pay
```

#### Запрос на оплату услуги Росреестра (по оферте):

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer  
testservice" -d '  
{  
  "orderID": "12345",  
  "depType": "rosreestrOffer",  
  "payType": "card",  
  "kvit": true,  
  "payerParams": {  
    "fio": "Иванов Иван Иванович"  
  },  
  "paymentParams": {  
    "supplierBillID": "32110238079253900800"  
  },  
  "returnUrl": "https://testpartner.ru/e0462fc0872e2/",  
  "sign": "7453035b6dd3107656c33ec337711999"  
}'  
  
https://demopay.***.ru/api/multiserv/2.0/pay
```

#### Запрос на оплату Пожертвования:

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer  
testservice" -d '  
{  
  "orderID": "12345",  
  "depType": "donation",  
  "payType": "card",  
  "kvit": false,  
  "amount": 10000,  
  "payerParams": "",  
}
```

```
"paymentParams": {
  "purpose": "Добровольное пожертвование"
},
"returnUrl": "https://testpartner.ru/e0462fc0872e2/",
"sign": "a7d6829e5a4e0a57e87dbe0adcba5208"
} '
https://demopay.***.ru/api/multiserv/2.0/pay
```

### Запрос на оплату судебной госпошлины:

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer
testservice" -d '
{
  "orderId": "6280536",
  "depType": "justice",
  "payType": "card",
  "kvit": true,
  "amount": 30000,
  "payerParams": {
    "fio": "Иванов Иван Иванович",
    "address": "г. Санкт-Петербург, ул. Оптиков, Дом 1, Корпус 2,
Квартира 3",
    "payerDoc": [
      {
        "code": "01",
        "value": ["1234567890"]
      }
    ]
  },
  "paymentParams": {
    "purpose": "Оплата пошлины, Иванов Иван Иванович, Паспорт РФ 1234567890",
    "kbk": "18210803010011000110",
    "oktmo": "40322000",
```

```
"inn": "7727406020",
"kpp": "770801001",
"bik": "017003983",
"correspondentBankAccount": "40102810445370000059",
"accountNumber": "03100643000000018500",
"payeeName": "Управление Федерального казначейства по Тульской области
(Межрегиональная инспекция Федеральной налоговой службы по управлению долгом)"
},
"sign": "b1ea695b777a9a0c8ce789570ff8f68b"
}'
https://demopay.***.ru/api/multiserv/2.0/pay
```

#### Запрос формирование выплаты на СБП:

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer
testservice" -d '
{
"orderID": "12345",
"depType": "nft7842113520",
"payType": " b2csbpay",
"kvit": false,
"amount": 5500,
"payerParams": {},
"paymentParams": {
"targetAccount": "9110000000"
"targetAdditional": "MemberID=123|Reason=40"
},
"returnUrl": "https://testpartner.ru/e0462fc0872e2/",
"sign": "d94c3157500e45ffa87261c05c64d258"
}'
https://demopay.***.ru/api/multiserv/2.0/pay
```

Запрос на оплату задолженности за проезд по дорогам с системой взимания оплаты «Свободный поток»:

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer testservice" -d '{
  "orderID": "6280536",
  "depType": "freeFlow",
  "payType": "card",
  "kvit": true,
  "amount": 6500,
  "payerParams": {
    "email": "ivanov@mail.ru",
    "phone": "9111234567"
  },
  "paymentParams": {
    "country": "RUS",
    "grnz": "B204EH37",
    "plazaTransactions": [395064880]
  },
  "sign": "b1ea695b777a9a0c8ce789570ff8f68b"
}'
https://demopay.***.ru/api/multiserv/2.0/pay
```

Запрос на оплату налогового начисления на ЕНС:

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer testservice" -d '{
  "orderID": "12345",
  "depType": "gibdd",
  "payType": "card",
  "kvit": true,
  "amount": 6500,
  "payerParams": {
```

```
"fio": "Иванов Иван Иванович",
"payerDoc": [
    {
        "code": "21",
        "value": ["123456789012"]
    }
]
},
"paymentParams": {
    "supplierBillID": "0"
},
"returnUrl": "https://testpartner.ru/e0462fc0872e2/",
"sign": "d94c3157500e45ffa87261c05c64d258"
}'

https://demopay.***.ru/api/multiserv/2.0/pay
```

#### Запрос перевода с СБП на СБП:

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer
testservice" -d '
{
"orderId": "12345",
"depType": " sbpToSbp1",
"payType": " sbp2Sbp",
"kvit": false,
"amount": 5500,
"payerParams": {},
"paymentParams": {
    "sbpPhone": "79191013456",
    "sbpBankID": "1000000000007",
    "sbpDescription": "Описание перевода"
},
"returnUrl": "https://testpartner.ru/e0462fc0872e2/",
```

```
"returnFailUrl": "https://testpartner.ru/fail/e0462fc0872e2/",  
"sign": "9515ef95049bec39698772a6dfbad05e"  
}'
```

https://demopay.\*\*\*.ru/api/multiserv/2.0/pay

#### Запрос оплаты NFT через СБП:

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer  
testservice" -d '
```

```
{  
  "orderID": "12345",  
  "depType": "nft7842113520",  
  "payType": "sbp",  
  "kvit": false,  
  "amount": 10000,  
  "payerParams": {},  
  "paymentParams": {},  
  "sign": "63971ddba4a569e5ade2e575713da9ce"  
}'
```

https://demopay.\*\*\*.ru/api/multiserv/2.0/pay

#### Запрос оплаты штрафа ГИБДД через СБП:

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer  
testservice" -d '
```

```
{  
  "orderID": "12345",  
  "depType": "gibdd",  
  "payType": "sbp",  
  "kvit": true,  
  "payerParams": {
```

```
"fio": "Иванов Иван Иванович"
},
"paymentParams": {
  "supplierBillID": "18810177170712879661"
},
"returnUrl": "https://testpartner.ru/e0462fc0872e2/",
"sign": "d94c3157500e45ffa87261c05c64d258"
}'
```

https://demopay.\*\*\*.ru/api/multiserv/2.0/pay

#### Запрос на пакетную оплату штрафов ГИБДД (по реквизитам):

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer
testservice" -d '
{
  "orderID": "6280536",
  "payType": "card",
  "kvit": false,
  "amount": 75000,
  "totalSum": 90000,
  "payerParams": {
    "fio": "Иванов Иван Иванович"
  },
  "paymentParams": {
    "packetAdvancePayments": [ {
      "supplierBillID": "18810510240731006620",
      "amount": 25000,
      "totalSum": 30000,
      "purpose": "Оплата штрафа по постановлению 18810510240731006620 от
31.07.2024, № от 31.07.2024",
      "kbk": "18811601121010001140",
      "oktmo": "86701000",
      "inn": "1001041280",
```

```
    "kpp": "100101001",
    "bik": "018602104",
    "correspondentBankAccount": "40102810945370000073",
    "accountNumber": "03100643000000010600",
    "payeeName": "УФК ПО РЕСПУБЛИКЕ КАРЕЛИЯ (МВД по Республике
Карелия)"
  },
  {
    "supplierBillID": "18810554240720092711",
    "amount": 50000,
    "totalSum": 60000,
    "purpose": "Оплата штрафа по постановлению 18810554240720092711 от
20.07.2024, № от 20.07.2024",
    "kbk": "18811601121010001140",
    "oktmo": "50701000",
    "inn": "5406012253",
    "kpp": "540601001",
    "bik": "015004950",
    "correspondentBankAccount": "40102810445370000043",
    "accountNumber": "03100643000000015100",
    "payeeName": "УФК по Новосибирской области (ГУ МВД РФ по
Новосибирской области)"
  }
]
},
"returnUrl": "https://testpartner.ru/e0462fc0872e2/",
"returnFailUrl": "https://testpartner.ru/2e0462fc0872e2/",
"sign": "903987da148b176376310de26746779f"
}

https://demopay.***.ru/api/multiserv/2.0/pay
```

Запрос на оплату начисления по платным дорогам из ГИС ГМП:

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer
testservice" -d '
{
"orderId": "12345",
"depType": " paidRoads",
"payType": "card",
"kvit": true,
"payerParams": {
  "fio": "Иванов Иван Иванович"
  "email": "ivanov@mail.ru"
},
"paymentParams": {
  "supplierBillID": " 0412479001000013586054242"
},
"returnUrl": "https://testpartner.ru/e0462fc0872e2/",
"sign": "d94c3157500e45ffa87261c05c64d258"
}'

https://demopay.***.ru/api/multiserv/2.0/pay
```

### Запрос на оплату через СБП с подпиской

#### Запрос:

```
{
"orderId": "12345674",
"depType": "gibdd",
"payType": "sbp",
"kvit": false,
"payerID": "745",
"sbpSubscription": {
  "enable": true,
  "purpose": "Ежемесячная подписка"
},
"payerParams": {
```

```
"fio": "Иванов Иван Иванович",
"email": "i.ivanov@example.ru"
},
"paymentParams": {
  "supplierBillID": "18810110160623813543"
},
"sign": "2f76a2e6a068820ad4557b7227d25d25"
}
```

Ответ:

```
{
  "paymentID": 511147
}
```

Ответ при оплате банковской картой:

```
{
  "paymentID": 511147,
  "redirectUrl": "https://testpayservice.ru/e0462fcfrgghf08gf72e2"
}
```

Ответ с 3ds (при оплате с криптограммой):

```
{
  "paymentID": 511147,
  "3ds":
}
```

Ответ для перевода с СБП на СБП:

```
{
  "paymentID": 511147,
  "payloadSbpToSbp": "https://qr.nspk.ru/AD10005F3ED9GCLM94UBH9UKCVN5V07
K?type=02&bank=100000000030&sum=1000&cur=RUB&crc=A479",
  "QRImgSbpToSbp": "base64_encoded_string"
}
```

```
}
```

### Ответ для оплаты штрафа через СБП:

```
{  
  "paymentID": 511147,  
  "sbpCode": "https://qr.nspk.ru/AD10005F3ED9GCLM94UBH9UKCVN5V07K?type=0  
2&bank=100000000030&sum=1000&cur=RUB&crc=A479",  
  "sbpCodeImageUrl": " https://qr.test.1mps.ru/image/00000c3558/?size=M"  
  "redirectUrl": "https://testpayservice.ru/e0462fcfrgghf08gf72e2"  
}
```

#### 2.2.2.3 Метод paymentInfo

Получение информации по Платежу.

Адрес запроса: GET

[https://demopay.\\*\\*\\*.ru/api/multiserv/2.0/paymentInfo/paymentID](https://demopay.***.ru/api/multiserv/2.0/paymentInfo/paymentID)

#### Схема взаимодействия

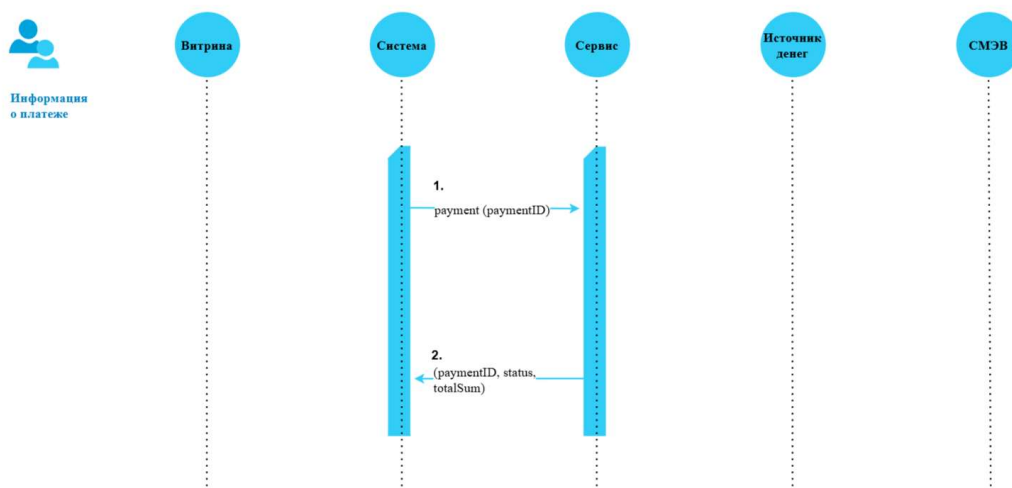


Рисунок 7. Схема взаимодействия при получении информации о платеже

#### Параметры

Входные параметры:

Отсутствуют.

Выходные параметры:

Параметр	Тип	Описание
paymentID	int	Идентификатор Платежа в ПЦ.
orderID	string	Уникальный номер Заказа на стороне Партнера.
uid	string	Уникальный Идентификатор Платежа. Для авторизованных платежей, импортированных ГИС ГМП.
importGGDate	string	Дата и время импорта факта оплаты в ГИС ГМП.
status	object	Статус платежа в ПЦ (см. подробнее тип statuses).
code	int	Код статуса Платежа.
name	string	Наименование статуса Платежа.
date	string	Время установки статуса Платежа, указывается только для конечных статусов (auth и cancel).
cancelReason	string	Описание причины отмены платежа, указывается только для статуса cancel. (см. 4.9).
depType	string	Тип Платежа (см. подробнее тип depType).
kvit	boolean	Заказ Квитанции, возможные значения: <ul style="list-style-type: none"> <li>• true – с квитанцией;</li> <li>• false – без квитанции.</li> </ul>
amount	int	Сумма Платежа в пользу Получателя, в копейках.
payType	string	Способ оплаты, возможные значения: <ul style="list-style-type: none"> <li>• phone - со счета мобильного телефона;</li> <li>• card – банковской картой;</li> <li>• sbr – через сервис быстрых платежей ЦБ РФ.</li> </ul>
paymentParams	object	Параметры Платежа, переданные в запросе pay.
linkCard	boolean	Признак заказа привязки БК на витрине.
card	object	Параметры карты, с которой был совершен платеж.

pan	string	Маскированный номер карты. Отображаются первые 6 и последние 4 цифры.
cardType	string	Тип банковской карты (VISA, MASTERCARD, MIR и т.д.)
cardId	string	Идентификатор привязанной карты.
result	object	Массив параметров Платежа (оферта).
sbpSubscription	object	Параметры подписки СБП.
subscriptionUuid	string	Идентификатор подписки СБП (который отправляется в рекуррентном платеже).
additionalData	object	Дополнительные параметры Платежа.

## Примеры

### Запрос (по paymentId):

```
curl -i -X GET -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer testservice" -d https://demopay.***.ru/api/multiserv/2.0/paymentInfo/1234567890
```

### Запрос (по orderId):

```
curl -i -X GET -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer testservice" -d https://demopay.***.ru/api/multiserv/2.0/paymentInfo/12345
```

### Ответ (базовый пример):

```
{
  "paymentID" : 1234567890,
  "orderId": "12345",
  "uip": "10445256781000012205201961760947",
  "importGGDate": "2017-05-02T10:12:25+03:00",
  "status" : {
    "code" : 20,
    "name" : "auth",
    "date" : "2017-05-02T10:11:37+03:00"
  }
}
```

```

    },
    "depType" : "gibdd",
    "kvit": true,
    "amount" : 25000,
    "totalSum" : 25375,
    "payType" : "card",
    "paymentParams" :
        {
            "supplierBillID":["18810107170424845551"]
        },
    "card":
    {
        "pan":"427432*****5579",
        "cardType": " VISA",
        "cardId": "123563861202"
    },
    "result":
    [
        "DepartmentalInfo.DrawerStatus" : "24",
        "DepartmentalInfo.CBC" : "18811630020016000140",
        "DepartmentalInfo.OKATO" : "83701000",
        "DepartmentalInfo.PaytReason" : "0",
        "DepartmentalInfo.TaxPeriod" : "0",
        "DepartmentalInfo.DocNo" : "24;0748823132",
        "DepartmentalInfo.DocDate" : "14.04.2017",
        "DepartmentalInfo.TaxPaytKind" : "0",
        "PayerName" : "ООО НКО «МОБИ.ДЕНЬГИ»//Иванов Иван Иванович//",
        "PayerInn" : "0",
        "PayerKPP" : "0",
        "PayeeName" : "УФК по Кабардино-Балкарской Республике {МВД по
КБР л\с 0404
1471170}",
        "PayeePersonalAcc" : "40101810100000010017",
        "PayeeBic" : "048327001",
        "PayeeInn" : "0711009240",

```

```
        "PayeeKPP" : "072501001",
        "Purpose" : "УИН18810107170424845551///ШТРАФ ПО
АДМИНИСТРАТИВНОМУ ПРАВОНАРУ
ШЕНИЮ, № 18810107170424845551 от 14.04.2017, Иванов Иван Иванович",
        "AccDocNo" : "589",
    ]
}
```

Ответ (с привязкой СБП):

```
{
  "paymentID": 115152412,
  "orderID": "234567002",
  "status": {
    "code": "20",
    "name": "auth",
    "date": "2025-01-20T15:24:49+03:00"
  },
  "depType": "fns",
  "kvit": true,
  "amount": 100,
  "totalSum": 110,
  "payType": "sbp",
  "paymentParams": {
    "supplierBillID": "18209966253930082696"
  },
  "result": [
    {
      "DepartmentalInfo.DrawerStatus": "20",
      "DepartmentalInfo.CBC": "18201061201010000510",
      "DepartmentalInfo.OKATO": "0",
      "DepartmentalInfo.PaytReason": "0",
      "DepartmentalInfo.TaxPeriod": "0",
      "DepartmentalInfo.DocNo": "0",
      "DepartmentalInfo.DocDate": "20.01.2025",
    }
  ]
}
```

```

        "DepartmentalInfo.TaxPaytKind": "0",
        "PayerName": "ООО НКО «МОБИ.ДЕНЬГИ»//Иванова Ирина
Владимировна//",
        "PayerInn": "781311034113",
        "PayerKPP": "0",
        "PayeeName": "Казначейство России (ФНС России)",
        "PayeePersonalAcc": "03100643000000018500",
        "PayeeCorrespAcc": "40102810445370000059",
        "PayeeBic": "017003983",
        "PayeeInn": "7727406020",
        "PayeeKPP": "770801001",
        "Purpose": "УИН18209966253930082696///Единый налоговый платеж,
Богатырева Ирина Вадимовна, 781311034893",
        "AccDocNo": "428"
    }
],
    "uip": "10445256781000012001202515152412",
    "sbpSubscription": {
        "subscriptionUuid": "01efd729-8b3d-6357-a8ff-000c29d59ec7"
    }
}

```

#### 2.2.2.4 Уведомление об успешном платеже notify

Уведомление отправляется на url с авторизационным токеном, которые должны быть предоставлены Партнером для отправки Сервисом уведомлений. Информация о Платеже идентична ответу на запрос paymentInfo + sign (формируется как строка md5 от конкатенации значений следующих параметров: токен партнера, paymentID, status.name; пример - a3b1f52f7b6c1d864ad7623023500f5a).

Доставка информации о конечном статусе платежа по notify не является гарантированной, для получения статуса платежа необходимо использовать метод paymentinfo.

Схема отправки уведомления, в случае не успешной доставки:

- сразу после наступления события;
- через 1 секунду;
- через 10 сек;

- через 1 мин;
- через 5 мин;
- через 20 мин;
- через 60 мин.

### 2.2.2.5 Метод kvit

Запрос на получение платежного поручения.

Адрес запроса: GET [https://demopay.\\*\\*\\*.ru/api/multiserv/2.0/kvit/paymentID](https://demopay.***.ru/api/multiserv/2.0/kvit/paymentID)

#### Схема взаимодействия

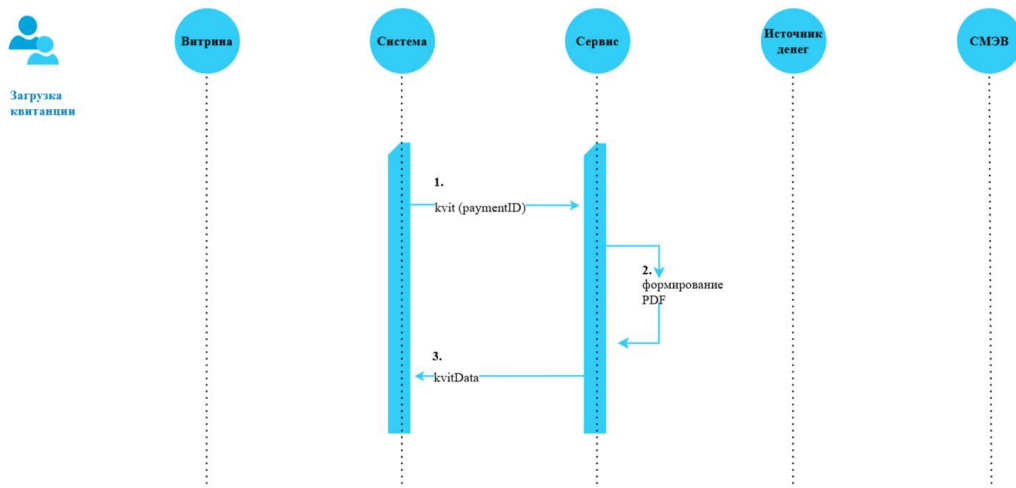


Рисунок 8. Схема загрузки квитанции

Параметры

Входные параметры:

Отсутствуют.

Выходные параметры:

Параметр	Тип	Описание
kvitData	array/string Base64	PDF документ с платежным поручением в формате ПД4. Платежное поручение предоставляется только по Платежам в статусе auth. <ul style="list-style-type: none"> <li>• string - для одного платежного поручения;</li> </ul>

		<ul style="list-style-type: none"> <li>• <code>array</code> - для нескольких платежных поручений при мультиплатеже.</li> </ul>
--	--	--

### **Ошибки:**

Код ошибки	Описание
100	Возникла ошибка при получении платежного поручения.
107	Запрошенный платеж не найден.
128	Отсутствуют права на предоставление платежного поручения по запрошенному платежу.
131	Запрошенный платеж не авторизован.
130	Запрос платежных поручений недоступен Партнёру. Обратитесь к менеджеру.
113	По указанному платежу отсутствует платежного поручения.

### Примеры

#### **Запрос:**

```
curl -i -X GET -H "Content-Type: application/json; charset=utf-8" -H
"Authorization: Bearer testservice" -
d https://demopay.***.ru/api/multiserv/2.0/kvit/1234567890
```

#### **Ответ (1 платежное поручение):**

```
{"kvitData": "base64_encoded_string"}
```

#### **Ответ (несколько платежных поручений):**

```
{"kvitData": ["base64_encoded_string", "base64_encoded_string"]}
```

## 2.3 Метод `balance`

Запрос на получение текущего баланса на счете обеспечения Партнера.

Адрес запроса: GET [https://demopay.\\*\\*\\*.ru/api/multiserv/2.0/balance](https://demopay.***.ru/api/multiserv/2.0/balance)

### Схема взаимодействия

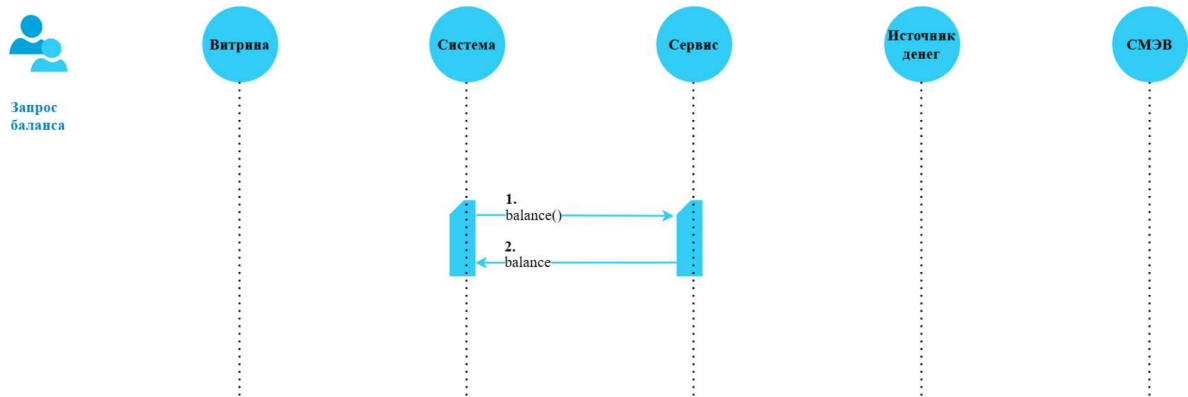


Рисунок 9. Схема запроса баланса

### Параметры

#### Входные параметры:

Отсутствуют.

#### Выходные параметры:

Параметр	Тип	Описание
balance	int	Сумма на счёте обеспечения Партнера, указывается в копейках.

### Примеры

#### Запрос:

```
curl -i -X GET -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer testservice" -d https://demopay.***.ru/api/multiserv/2.0/balance
```

#### Ответ:

```
{"balance": 100000}
```

### 2.3.1.1 Метод linkCard

Запрос на привязку банковской карты Пользователя по результатам успешного платежа.

Адрес запроса: POST [https://demopay.\\*\\*\\*.ru/api/multiserv/2.0/linkCard](https://demopay.***.ru/api/multiserv/2.0/linkCard)

#### Схема взаимодействия

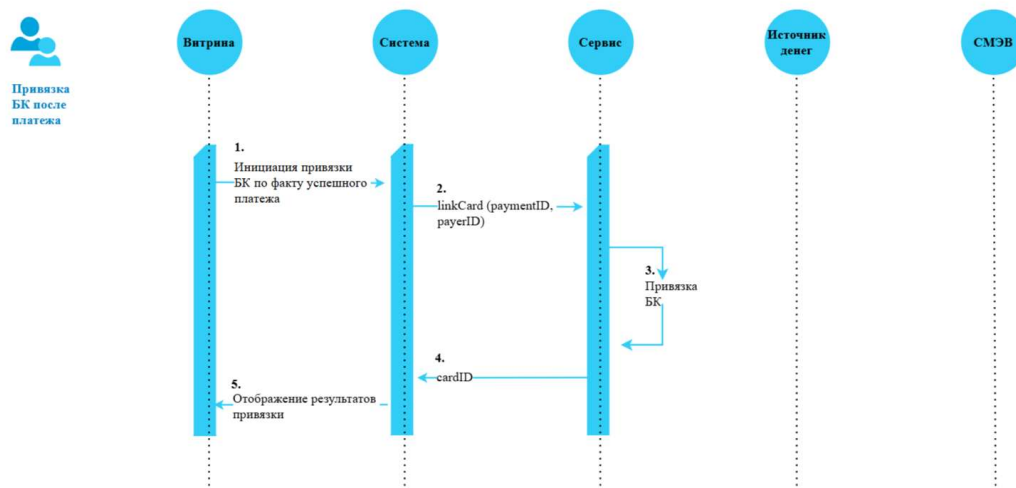


Рисунок 10. Схема привязки БК после платежа

#### Параметры

##### Входные параметры:

Параметр	Тип	Об.	Описание
paymentID	int	1	Идентификатор Платежа в ПЦ.
payerID	string	1	Длина 28 символов (латинских букв/цифр). Идентификатор Пользователя в информационной системе Партнера. Значение должно быть уникальным для каждого Пользователя. Параметр обязателен для заполнения при оплате с Банковской карты.
cardName	string	0..1	Наименование карты, которое будет отображаться Пользователю при оплате с привязанной карты на странице эквайринга.

## Выходные параметры:

Параметр	Тип	Описание
cardID	string	Идентификатор привязанной к payerID банковской карты.

Примеры

## Запрос:

```
curl -i -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer testservice" -d '{"paymentID": "1234567890", "payerID": "user11111122233334456782345", "cardName": "дебетовая карта"}' https://demopay.***.ru/api/multiserv/2.0/linkCard
```

## Ответ:

```
{"cardID": "123"}
```

### 2.3.1.2 Метод unlinkCard

Запрос на удаление привязки банковской карты Пользователя.

Адрес запроса: POST [https://demopay.\\*\\*\\*.ru/api/multiserv/2.0/unlinkCard](https://demopay.***.ru/api/multiserv/2.0/unlinkCard)

## Схема взаимодействия

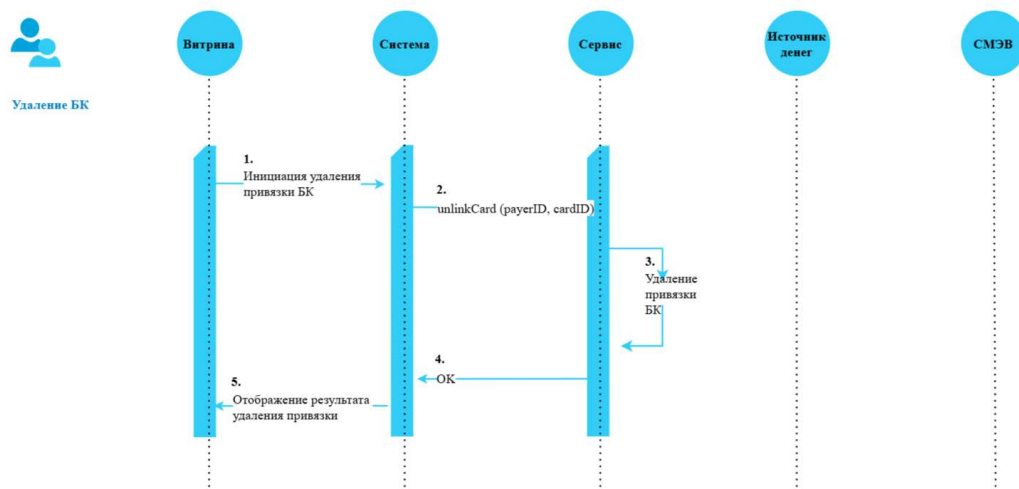


Рисунок 11. Схема удаления БК

## Параметры

### Входные параметры:

Параметр	Тип	Об.	Описание
payerID	string	1	Длина 28 символов (латинских букв/цифр). Идентификатор Пользователя в информационной системе Партнера.
cardID	string	1	Идентификатор привязанной к payerID банковской карты.
depType	string	1	Наименование ведомства Получателя.

### Выходные параметры:

Отсутствуют.

## Примеры

### Запрос:

```
curl -i -X POST -H "Content-Type: application/json; charset=utf-8" -H
"Authorization: Bearer testservice" -d
'{"payerID":"user111111122233334456782345","cardID":"123","depType":"fns"
}'
https://demopay.***.ru/api/multiserv/2.0/unlinkCard
```

### 2.3.1.3 Метод cards

Запрос на получение списка привязанных банковских карт Пользователя.

Адрес запроса: GET [https://demopay.\\*\\*\\*.ru/api/multiserv/2.0/cards/payerID](https://demopay.***.ru/api/multiserv/2.0/cards/payerID)

## Схема взаимодействия

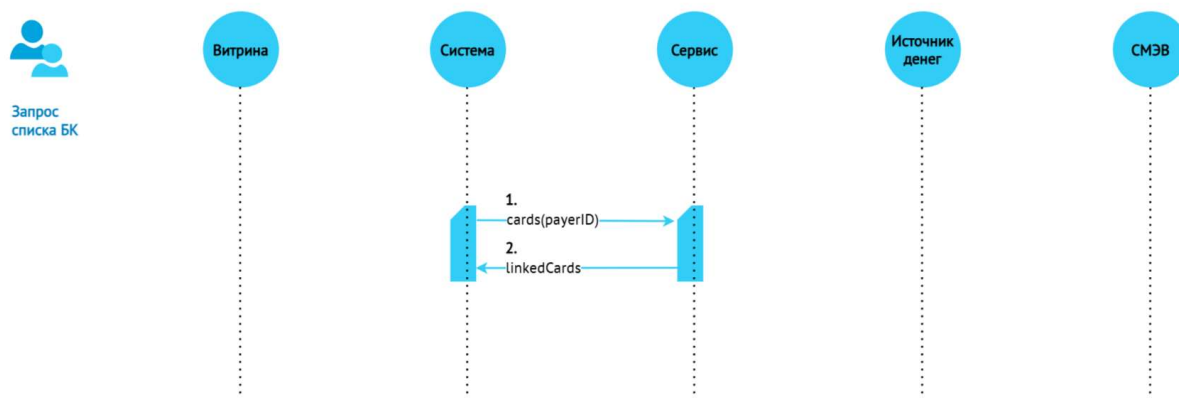


Рисунок 12. Схема запроса списка БК

## Параметры

### Входные параметры:

Отсутствуют

### Выходные параметры:

Параметр	Тип	Об.	Описание
payerID	string	1	Длина 28 символов (латинских букв/цифр). Идентификатор Пользователя в информационной системе Партнера.
linkedCards	array	0..1	Массив привязанных банковских карт с данными по каждой карте.
cardID	string	1	Идентификатор привязанной к payerID банковской карты.
cardName	string	0..1	Наименование карты, которое отображается Пользователю при оплате с привязанной карты на странице эквайринга.
pan	string	1	Маскированный номер привязанной карты.
cardType	string	1	Тип банковской карты (VISA, MASTERCARD, MIR и т.д.)

expired	int	1	Признак истечения срока действия карты: 0 – срок действия не истек, 1 – карта просрочена.
---------	-----	---	---

## Примеры

### Запрос:

```
curl -i -X GET -H "Content-Type: application/json; charset=utf-8" -H
"Authorization: Bearer testservice" -
d https://demopay.***.ru/api/multiserv/2.0/cards/user12345
```

### Ответ:

```
{
  "payerID": "user111111122233334456782345",
  "linkedCards": [
    {
      "cardID": "402785202",
      "pan": "512303*****0081",
      "cardType": "MASTERCARD",
      "expired": 1,
      "cardName": "кредитка"
    }
  ]
}
```

### 2.3.1.4 Метод recurrent

Инициация оплаты по Начислению или Авансового платежа без участия Пользователя с ранее привязанной банковской карты или через систему СБП. Использование метода возможно только после согласования с менеджером.

Адрес запроса: POST [https://demopay.\\*\\*\\*.ru/api/multiserv/2.0/recurrent](https://demopay.***.ru/api/multiserv/2.0/recurrent)

## Параметры

### Входные параметры:

Название	Значение	Об.	Описание
orderID	string	1	Уникальный номер заказа на стороне Партнера
payerID	string	1	Длина 28 символов (латинских букв/цифр). Идентификатор Пользователя в

			информационной системе Партнера. Значение должно быть уникальным для каждого Пользователя.
cardID	int	0	Идентификатор привязанной к payerID банковской карты
depType	string	1	Наименование ведомства Получателя
payType	string	0..1	Способ оплаты, возможные значения: <ul style="list-style-type: none"> <li>• phone – со счета мобильного телефона;</li> <li>• card – банковской картой;</li> <li>• deposit – со счета обеспечения Партнера;</li> <li>• pscb – со счета обеспечения в АО “ПСКБ”;</li> <li>• sbp – через сервис быстрых платежей ЦБ РФ;</li> <li>• b2csbpay – выплата на СБП со счета обеспечения Партнера;</li> <li>• sbp2sbp – выплата с СБП на СБП для ФЛ.</li> </ul>
kvit	boolean	1	Заказ квитанции, возможные значения: <ul style="list-style-type: none"> <li>• true – с квитанцией;</li> <li>• false – без квитанции.</li> </ul>
sbpSubscription	object	0..1	Запрос рекуррентного платежа по СБП.
subscriptionUuid	string	0..1	Идентификатор рекуррентного платежа.
amount	int	0..1	Сумма платежа без учета комиссий, обязательна для Авансовых платежей и при частичной оплате Начисления, указывается в копейках
totalSum	int	0..1	Сумма платежа с учетом комиссий, которая будет списана с Пользователя, в копейках, используется при расчете суммы списания с Пользователя ИС Партнера, использование параметра возможно только после

			согласования с менеджером. В случае использования с мультиплатежами должна быть указана для каждой операции.
payerParams	object	1	Данные о Пользователе (см. подробнее тип payerParams)
paymentParams	object	1	Данные по оплачиваемому Начислению и Получателю платежа (см. подробнее тип paymentParams)
sign	string	1	Подпись запроса. Формируется в зависимости от платежа. Подробнее см. в разделе Подпись (sign).  Секретное слово индивидуально для каждого Партнера.

#### Выходные параметры:

Название	Значение	Описание
paymentID	int	Идентификатор Платежа в ПЦ

#### **Примеры**

##### Запрос на рекуррентную оплату начисления

##### Запрос:

POST http://pay.local/api/multiserv/2.0/recurrent

Content-Type: application/json

Authorization: Bearer CRYPTO

```
{
  "orderID": "12345674",
  "depType": "gibdd",
  "payType": "sbp",
  "kvit": false,
  "sbpSubscription": {
    "subscriptionUuid": "01efacd7-4a54-691f-8e43-0242ac1e000a"
```

```
    },
    "payerParams": {
      "fio": "ОКс ОКс",
      "email": "serbinova@tprs.ru"
    },
    "paymentParams": {
      "supplierBillID": "18810110160623813543"
    },
    "sign": "1463ce4cc802959ae90fcaac879fea14"
  }

https://demopay.***.ru/api/multiserv/2.0/recurrent
```

Ответ:

```
{
  "paymentID": 511147
}
```

### 2.3.1.5 Метод saveCard

Запрос на сохранение карты пользователя, через временную блокировку проверочной суммы на карте (до 10 руб.). Используется только совместно с методом confirmSave.

Адрес запроса: POST [https://demopay.\\*\\*\\*.ru/api/multiserv/2.0/saveCard](https://demopay.***.ru/api/multiserv/2.0/saveCard)

### Схема взаимодействия

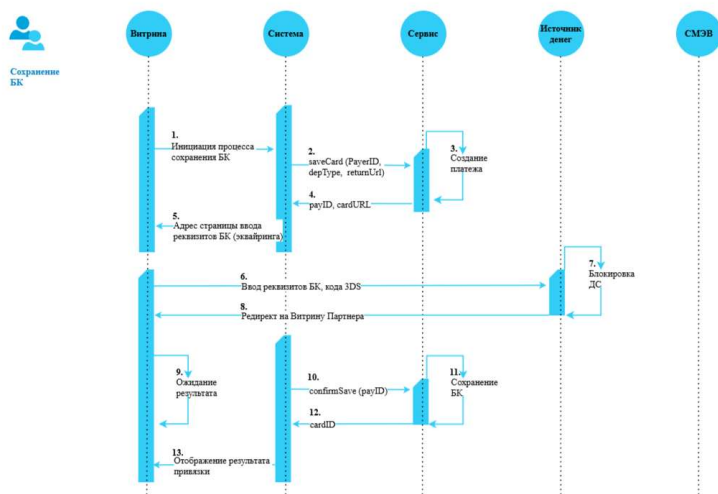


Рисунок 13. Схема взаимодействия при сохранении карты

## Параметры

### Входные параметры:

Название	Значение	Об.	Описание
payerID	string	1	Длина 28 символов (латинских букв/цифр). Идентификатор Пользователя в информационной системе Партнера. Значение должно быть уникальным для каждого Пользователя.
depType	string	1	Наименование ведомства Получателя.
returnUrl	string	1	Адрес Витрины для возврата Пользователя после сохранения карты.

### Выходные параметры:

Название	Значение	Об.	Описание
payID	string	1	Идентификатор проверочного платежа, необходимый для подтверждения сохранения карты (входной параметр для метода confirmSave).
cardUrl	string	1	Адрес для ввода Пользователем реквизитов банковской карты.

## Примеры

### Запрос:

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer testservice" -d '{
  "payerID": "user111111122233334456782345",
  "depType": "gibdd",
  "returnUrl": "https://testpartner.ru/e0462fc0872e2/"
}'
https://demopay.***.ru/api/multiserv/2.0/saveCard
```

### Ответ:

```
{
  "payID": "511147",
  "cardUrl": "https://testpartner.ru/e0462fc0872e2/"
}
```

### 2.3.1.6 Метод confirmSave

Запрос на подтверждения сохранения карты.

Используется только совместно с методом saveCard

Адрес запроса: POST [https://demopay.\\*\\*\\*.ru/api/multiserv/2.0/confirmSave](https://demopay.***.ru/api/multiserv/2.0/confirmSave)

## Параметры

### Входные параметры:

Название	Значение	Об.	Описание
payID	string	1	Идентификатор проверочного платежа, переданный в выходных параметрах метода saveCard.
depType	string	1	Наименование ведомства Получателя.

### Выходные параметры

Название	Значение	Об.	Описание
----------	----------	-----	----------

cardID	string	1	Идентификатор привязанной к payerID банковской карты.
--------	--------	---	---

## Примеры

### Запрос:

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer testservice" -d '{
  "payID": "511147",
  "depType": "gibdd"
}'
https://demopay.***.ru/api/multiserv/2.0/confirmSave
```

### Ответ:

```
{
  "cardID": "12312347"
}
```

### 2.3.1.7 Метод searchPaid

Запрос оплаченных Начислений по идентификаторам Пользователя или УИН. Поиск Начислений осуществляется асинхронно через среду СМЭВ в ГИС ГМП и информационных системах государственных ведомств.

Для получения полных данных по результатам поиска от всех источников необходимо повторять запрос до получения в ответе параметра status = complete.

Адрес запроса: POST [https://demopay.\\*\\*\\*.ru/api/multiserv/2.0/searchPaid](https://demopay.***.ru/api/multiserv/2.0/searchPaid)

## Схема взаимодействия

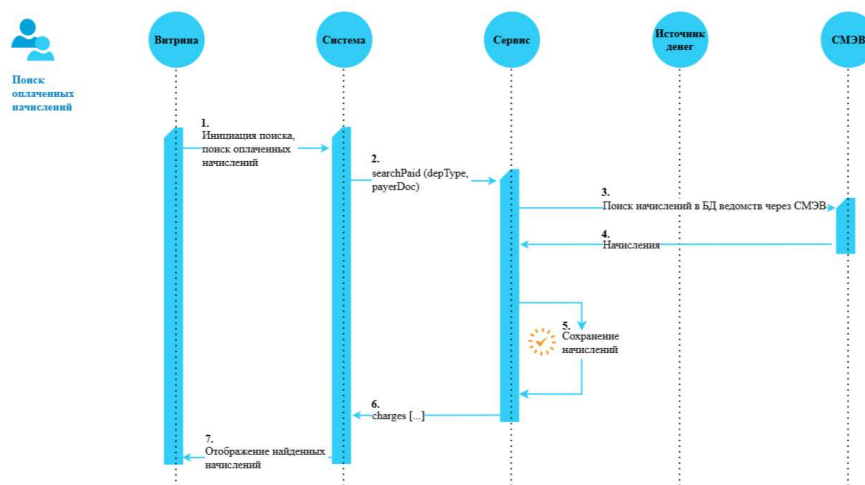


Рисунок 14. Схема поиска оплаченных начислений

## Параметры

### Входные параметры:

Название	Значение	Об.	Описание
depType	string	1	Определяет тип ведомства поиска (см. тип depType).
payerDoc	array	1	Идентификаторы Пользователя для осуществления поиска Начислений (см. подробнее тип payerDoc), кроме кодов 51, 91, 92, 93, 99, 931.

### Выходные параметры:

Название	Значение	Описание
status	string	Статус выполнения поиска по переданным идентификаторам: <ul style="list-style-type: none"> <li>part – частичный результат, ответ получен не от всех источников;</li> <li>complete – полный результат, поиск завершен.</li> </ul>
charges	array	Массив найденных начислений (см. подробнее тип charge).

## Примеры

### Запрос:

```
curl -i -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer testservice" -d '{"depType": "gibdd", "payerDoc": [{"code": "22", "value": ["7712345678", "7712345679"]}]} ' https://demopay.***.ru/api/multiserv/2.0/searchPaid
```

Ответ:

```
{
  "status": "complete",
  "charges": [
    {
      "payerDoc": [ {"code": "22", "value": ["7712345678"]} ],
      "depType": "gibdd",
      "supplierBillID": "18810028150001578060",
      "chargeData":
        {
          "billDate": "2014-10-23T00:00:00Z",
          "amountToPay": 50000,
          "purpose": "ШТРАФ ПО АДМИНИСТРАТИВНОМУ ПРАВОНАРУШЕНИЮ",
          "kbk": "18811630020016000140",
          "oktmo": "45379000",
          "inn": "7707089101",
          "kpp": "770731005",
          "bik": "044583001",
          "accountNumber": "40101810800000010041",
          "payeeName": "ЦАФАП ОДД ГИБДД ГУ МВД России по г. Москве",
          "divID": 12345,
          "amount": 50000,
          "treasureBranch": "УФК по Московской области"
          "additionalData":
            {
              "discountSize": "50",
              "discountDate": "2016-10-26"
            },
        },
    },
  ],
}
```

```

    "paymentData":
    {
        "sum":"25000",
        "date":"2016-10-23T18:00:00Z"
    },
    "ttl":"2017-02-25T21:00:00Z"
}
]
}

```

### 2.3.1.8 Метод `getPaymentOrder`

Запрос платежного поручения в формате pdf. Платежное поручение формируется в течение 2 рабочих дней после оплаты начисления. Запрос рекомендуется направлять по прошествии указанного времени. Доступ к методу предоставляется по согласованию с менеджером.

Адрес запроса:

GET [https://demopay.\\*\\*\\*.ru/api/multiserv/2.0/getPaymentOrder/paymentID](https://demopay.***.ru/api/multiserv/2.0/getPaymentOrder/paymentID)

#### Схема взаимодействия

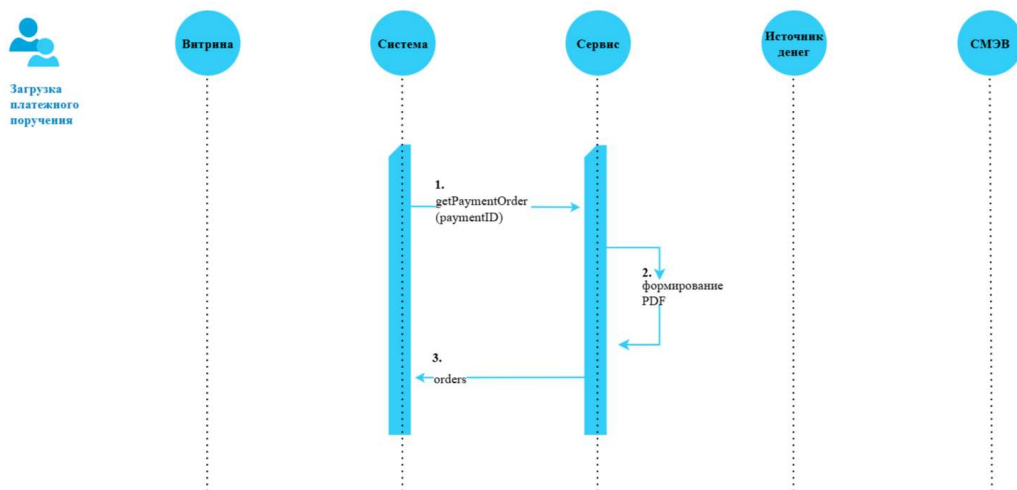


Рисунок 15. Схема загрузки платежного поручения

Параметры

**Входные параметры:**

Отсутствуют.

**Выходные параметры:**

Название	Значение	Описание
orders	array	Массив платежных поручений.
paymentOrder	String Base64	PDF-документ платежного поручения формы 0401060.

Ошибки:

Код ошибки	Описание
107	Запрошенный платеж не найден.
128	Отсутствуют права на предоставление платежного поручения по запрошенному платежу.
131	Запрошенный платеж не авторизован.
130	Запрос платежных поручений недоступен Партнёру. Обратитесь к менеджеру.
100	Возникла ошибка при получении ПП.

Примеры

Запрос:

```
curl -X GET --location "https://demopay.***.ru/api/multiserv/2.0/getPaymentOrder/1234567890" \
  -H "Content-Type: application/json; charset=utf-8" \
  -H "Authorization: Bearer testservice"
```

Ответ:

```
{
  "orders": [
    {
      "paymentOrder": "base64_encoded_string"
    }
  ]
}
```

```

    }
  ]
}

```

### 2.3.1.9 Метод getInn

Запрос ИНН физического лица. Запрос на получение ИНН осуществляется асинхронно через среду СМЭВ в УФНС.

Для получения ИНН необходимо повторять запрос до получения в ответе параметра status = complete.

Адрес запроса: POST https://demoray.\*\*\*.ru/api/multiserv/2.0/getInn

#### Схема взаимодействия

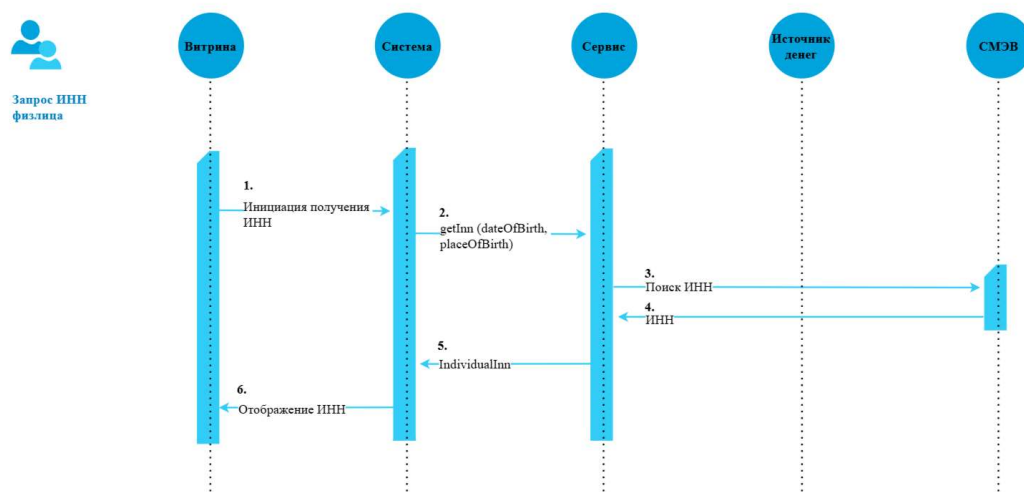


Рисунок 16. Схема запроса ИНН физ. лица

#### Параметры

##### Входные параметры:

Название	Значение	Об.	Описание
birth_date	string	1	Дата рождения физ. лица в формате ГГГГ-ММ-ДД.
fio	object	1	Сведения о ФИО физ. лица.
second_name	string	1	Фамилия физ. лица.

first_name	string	1	Имя физ. лица.
patronymic	string	0..1	Отчество физ. лица.
doc	object	1	Сведения о документе, удостоверяющем личность физ. лица
code	string	1	Код вида документа, удостоверяющего личность. Значения: <ul style="list-style-type: none"> <li>• 01 - Паспорт гражданина СССР;</li> <li>• 03 - Свидетельство о рождении;</li> <li>• 10 - Паспорт иностранного гражданина;</li> <li>• 12 - Вид на жительство в Российской Федерации;</li> <li>• 15 - Разрешение на временное проживание в Российской Федерации;</li> <li>• 19 - Свидетельство о предоставлении временного убежища на территории Российской Федерации;</li> <li>• 21 - Паспорт гражданина Российской Федерации;</li> <li>• 62 - Вид на жительство иностранного гражданина.</li> </ul>
number	string	1	Серия и номер документа, удостоверяющего личность. Паспорт гражданина Российской Федерации в формате ЧЧ ЧЧ ЧЧЧЧЧЧ.

Выходные параметры:

Название	Значение	Об.	Описание
status	string	1	Статус выполнения запроса: <ul style="list-style-type: none"> <li>• part – выполнение запроса не завершено;</li> <li>• complete – выполнение запроса завершено.</li> </ul>
individualInn	string	0..1	Иин физ. лица, отсутствует для статуса part. Для статуса complete отсутствует в случае

			непредоставления поставщиком сведений (ФНС) информации об ИНН ФЛ по переданным исходным данным.
--	--	--	---

## Примеры

### Запрос:

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer testservice" -d '{
  "birth_date": "1954-07-12",
  "fio": {
    "second_name": "Иван",
    "first_name": "Иванов",
    "patronymic": "Иванович"
  },
  "doc": {
    "code": "21",
    "number": "22 02 919928"
  }
}'
https://demopay.***.ru/api/multiserv/2.0/getInn
```

### Ответ:

```
{ "status": "complete",
  "individualInn": "111222333444" }
```

### 2.3.1.10 Уведомление о начислении по подписке subscribeNotify

Массив уведомлений по новым/измененным/оплаченным/аннулированным Начислениям, найденным по Идентификаторам Пользователя. Для активации подписки Партнеру необходимо предоставить менеджеру список типов уведомлений, которые планируется получать и url для отправки Сервисом уведомлений.

Для внесения в список вайтлист Партнеру необходимо запросить у менеджера список IP-адресов, с которых осуществляется отправка уведомлений.

Схема отправки уведомления, в случае не успешной доставки:

- сразу после наступления события;
- через 30 секунд после предыдущей попытки;
- через 1 минуту после предыдущей попытки;
- через 5 минут после предыдущей попытки;
- через 20 минут после предыдущей попытки;
- через 60 минут после предыдущей попытки.

### Схема взаимодействия

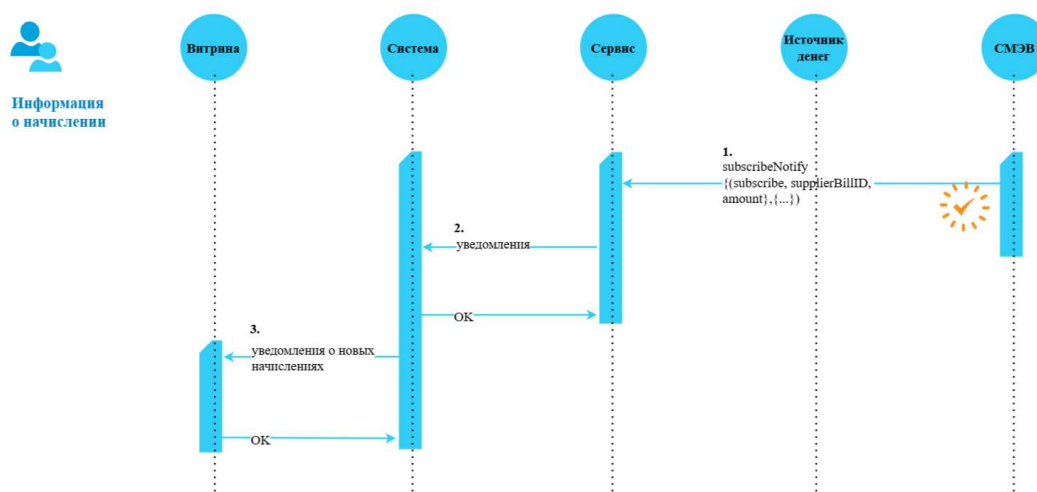


Рисунок 17. Схема формирования уведомления о начислении

### Параметры

Название	Значение	Об.	Описание
subscribe	string	1	Уникальный идентификатор Подписки Пользователя.
msgType	int	1	Тип уведомления, возможные значения: <ul style="list-style-type: none"> <li>• 1 – новое начисление;</li> </ul>

			<ul style="list-style-type: none"> <li>• 2 – уведомление об изменении начисления;</li> <li>• 3 – начисление аннулировано;</li> <li>• 4 – начисление оплачено.</li> </ul>
payerDoc	object	1	Идентификатор Пользователя, по которому найдено начисление (см. подробнее тип <code>payerDoc</code> ).
depType	string	1	Тип ведомства (см. подробнее тип <code>depType</code> ).
supplierBillID	string	1	Уникальный идентификатор Начисления.
billDate	string	1	Дата выставления Начисления.
amountToPay	int	0..1	Сумма к оплате в выставленном Начислении, указывается в копейках.
purpose	string	1	Назначение Платежа.
kbk	string	1	КБК.
payeeName	string	1	Наименование Получателя.
amount	int	1	Сумма начисления, указывается в копейках.
payerName	string	0..1	Наименование Плательщика по начислению (ФИО).
additionalDataDiscountSize	string	0..1	Размер скидки, в %.
additionalDataDiscountDate	string	0..1	Срок действия скидки.

## Примеры

### Запрос (HTTP):

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer testtoken" -d '
```

```

{
  [
    {
      "subscribe": "user12345",
      "msgType": 1,
      "payerDoc": {"code": "22", "value": "7712345678"},
      "depType": "gibdd",
      "supplierBillID ": "18810028150001578060",
      "billDate": "2014-10-23T00:00:00Z",
      "amountToPay": 50000,
      "purpose": "ШТРАФ ПО АДМИНИСТРАТИВНОМУ ПРАВОНАРУШЕНИЮ",
      "kbk": "18811630020016000140",
      "payeeName": "ЦАФАП ОДД ГИБДД ГУ МВД России по г. Москве",
      "amount": 50000,
      "payerName": "Иванов Иван Иванович",
      "additionalDataDiscountSize": "50",
      "additionalDataDiscountDate": "2016-10-20"
    },
    {"subscribe": "user12345",
      "msgType": 2,
      "payerDoc": {"code": "200", "value": "7717751677771801001"},
      "depType": "gibdd",
      "supplierBillID ": "18810550220130045532",
      "billDate": "2021-10-23T00:00:00Z",
      "amountToPay": 50000,
      "purpose": "ШТРАФ ПО АДМИНИСТРАТИВНОМУ ПРАВОНАРУШЕНИЮ",
      "kbk": "18811630020016000140",
      "payeeName": "ЦАФАП ОДД ГИБДД ГУ МВД России по г. Москве",
      "amount": 50000,
      "payerName": "ООО «Ромашка»",
      "additionalDataDiscountSize": "50",
      "additionalDataDiscountDate": "2021-11-10"},
    {...}
  ]
}

```

### **Ответ (HTTP):**

200 OK

#### **2.3.1.11 Метод subscribe**

Создание подписки на Уведомление по новым/измененным/оплаченным/аннулированным Начислениям.

Адрес запроса: POST [https://demopay.\\*\\*\\*.ru/api/multiserv/2.0/subscribe](https://demopay.***.ru/api/multiserv/2.0/subscribe)

## Схема взаимодействия

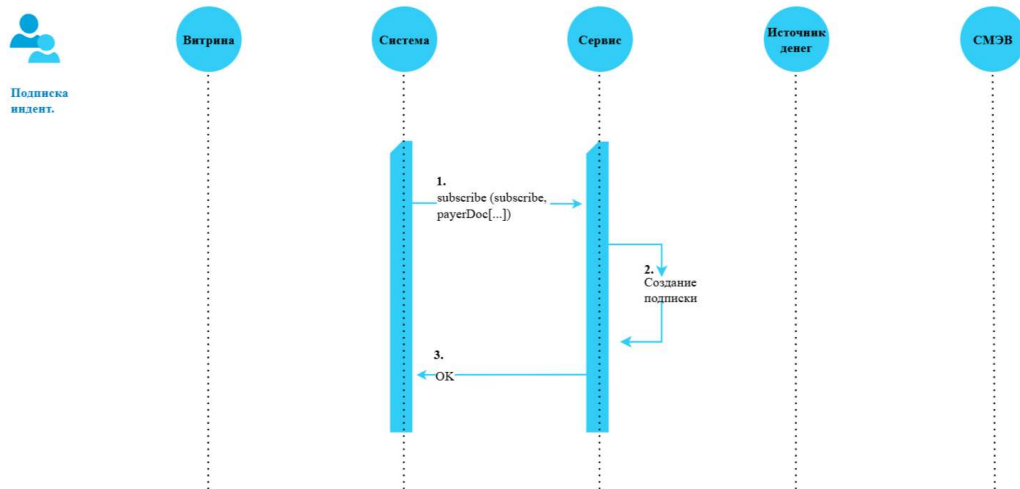


Рисунок 18. Схема создания подписки на уведомления по начислениям (новым, измененным, оплаченным, аннулированным)

## Параметры

### Входные параметры:

Название	Значение	Об.	Описание
subscribe	string	1	Уникальный идентификатор Подписки Пользователя.
payerDoc	array	1	Идентификаторы Пользователя для осуществления поиска Начислений (см. подробнее тип <code>payerDoc</code> ).

### Выходные параметры:

Название	Значение	Об.	Описание
status	string	1	Статус добавления Идентификаторов Пользователя в Подписку: 0 – успех, 1 – ошибка.

## Примеры

### Запрос:

```
curl -i -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer testservice" -d '{"subscribe": "user12345", "payerDoc":
```

```
[{"code": "22","value": ["7712345678","7712345679"]}, {"code": "202","value": ["7717751677771801001"]} ]' https://demopay.***.ru/api/multiserv/2.0/subscribe
```

**Ответ:**

```
{"status":"0"}
```

**2.3.1.12 Метод unsubscribe**

Удаление списка Идентификаторов Пользователя из Подписки.

Адрес запроса: POST https://demopay.\*\*\*.ru/api/multiserv/2.0/unsubscribe

**Схема взаимодействия**

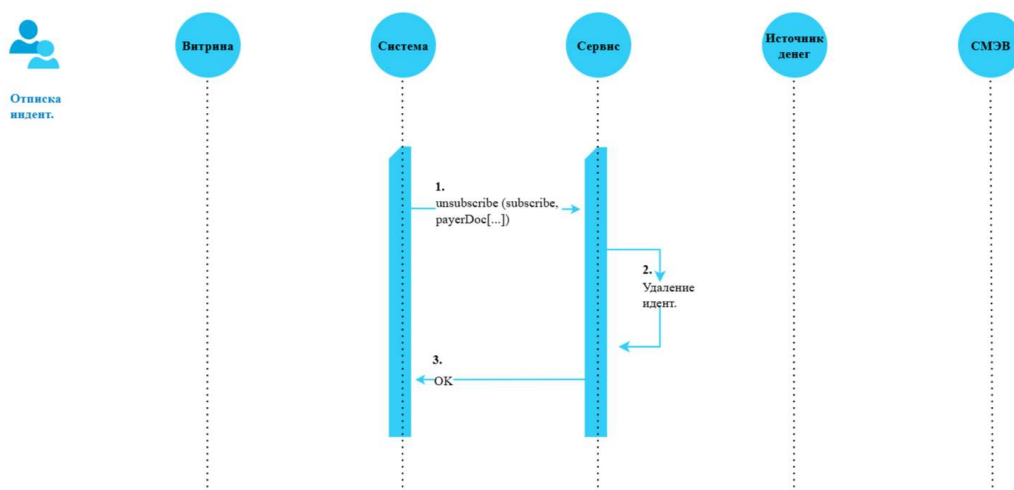


Рисунок 19. Схема удаления подписки на уведомления по начислениям (новым, измененным, оплаченным, аннулированным)

**Параметры**

**Входные параметры:**

Название	Значение	Об.	Описание
subscribe	string	1	Уникальный идентификатор Подписки Пользователя.
payerDoc	array	0..1	Идентификаторы Пользователя, которые необходимо удалить из подписки (см. подробнее

			тип payerDoc). Если payerDoc не передан, то удаляются все идентификаторы из подписки.
--	--	--	---

**Выходные параметры:**

Название	Значение	Об.	Описание
status	string	1	Статус удаления Идентификаторов Пользователя из Подписки: <ul style="list-style-type: none"> <li>• 0 – успех;</li> <li>• 1 – ошибка.</li> </ul>

Примеры

**Запрос:**

```
curl -i -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer testservice" -d '{"subscribe": "user12345", "payerDoc": [{"code": "22", "value": ["7712345678", "7712345679"]}]}'
```

[https://demopay.\\*\\*\\*.ru/api/multiserv/2.0/unsubscribe](https://demopay.***.ru/api/multiserv/2.0/unsubscribe)

**Ответ:**

```
{"status": "0"}
```

**2.3.1.13 Метод getSubscribeList**

Запрос списка Идентификаторов Пользователя из Подписки.

Адрес запроса:

GET [https://demopay.\\*\\*\\*.ru/api/multiserv/2.0/getSubscribeList/subscribe](https://demopay.***.ru/api/multiserv/2.0/getSubscribeList/subscribe)

**Схема взаимодействия**

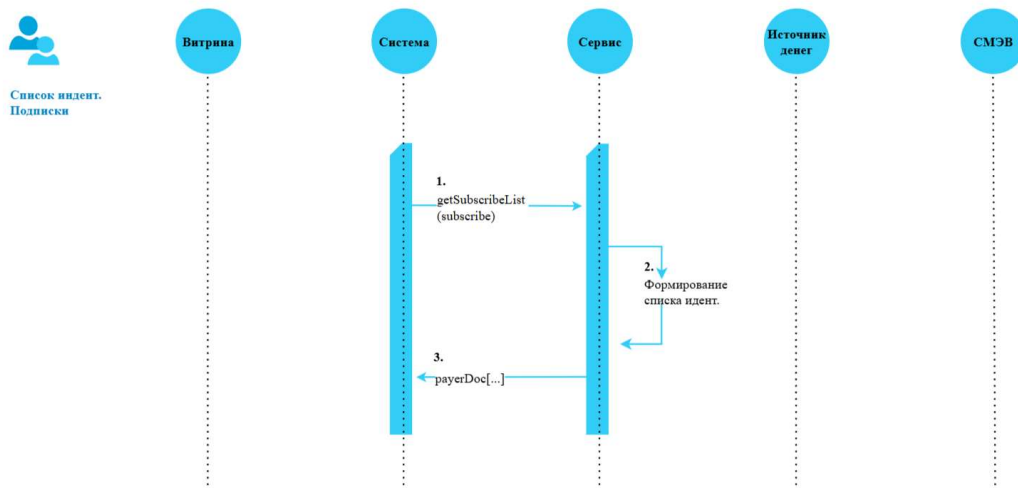


Рисунок 20. Схема получения списка идентификаторов пользователя в подписке

Параметры

**Входные параметры:**

Отсутствуют.

**Выходные параметры:**

Название	Значение	Об.	Описание
payerDoc	array	1	Идентификаторы Пользователя для осуществления поиска Начислений (см. подробнее тип <u>payerDoc</u> ).

Примеры

**Запрос:**

```
curl -i -X GET -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer testservice" -d https://demopay.***.ru/api/multiserv/2.0/getSubscribeList/subscribe12345
```

**Ответ:**

```
{"payerDoc": [ {"code": "22","value": ["7712345678"]}, {"code": "200","value": ["7717751677771801001"]}]}
```

### 2.3.1.14 Метод photo

Запрос на получение материалов фотофиксации административных правонарушений.

Адрес запроса: POST [https://demoray.\\*\\*\\*.ru/api/multiserv/2.0/photo](https://demoray.***.ru/api/multiserv/2.0/photo)

#### Схема взаимодействия

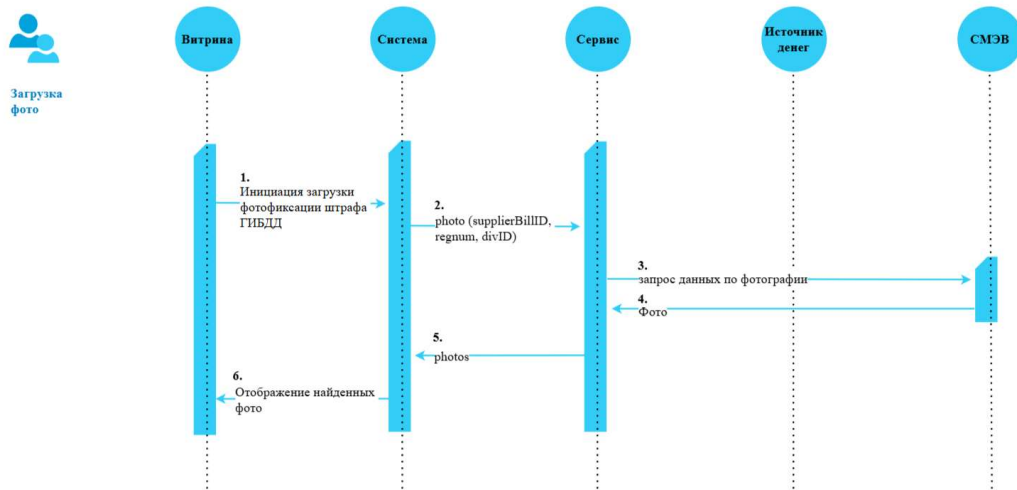


Рисунок 21. Схема загрузки фотофиксации штрафа

#### Параметры

##### Входные параметры:

Название	Значение	Об.	Описание
supplierBillID	string	1	Идентификатор Начисления.
regnum	string	1	Регистрационный номер транспортного средства, обязателен для получения фото.
sts	string	1	Номер свидетельства о регистрации транспортного средства, обязателен для получения фото.

##### Выходные параметры:

Параметр	Значение	Описание
photos	array	Массив из base64-кодированных снимков (.jpeg).

## Примеры

### Запрос:

```
curl -i -X POST -H "Content-Type: application/json; charset=utf-8" -H
"Authorization: Bearer testservice" -d
'{"supplierBillID":"18810107170424845551","regnum":
"A001AA78","sts": 7812345678}' https://demopay.***.ru/api/multiserv/2.0/photo
```

### Ответ:

```
{"photos":[
  "base64_encoded_photo1",
  "base64_encoded_photo2",
  "base64_encoded_photo3" ]}
```

### 2.3.1.15 Метод getSBPBankList

Запрос на получение списка банков, привязанных к СБП.

Адрес запроса: POST [https://demopay.\\*\\*\\*.ru/api/multiserv/2.0/getSBPBankList](https://demopay.***.ru/api/multiserv/2.0/getSBPBankList)

### Схема взаимодействия

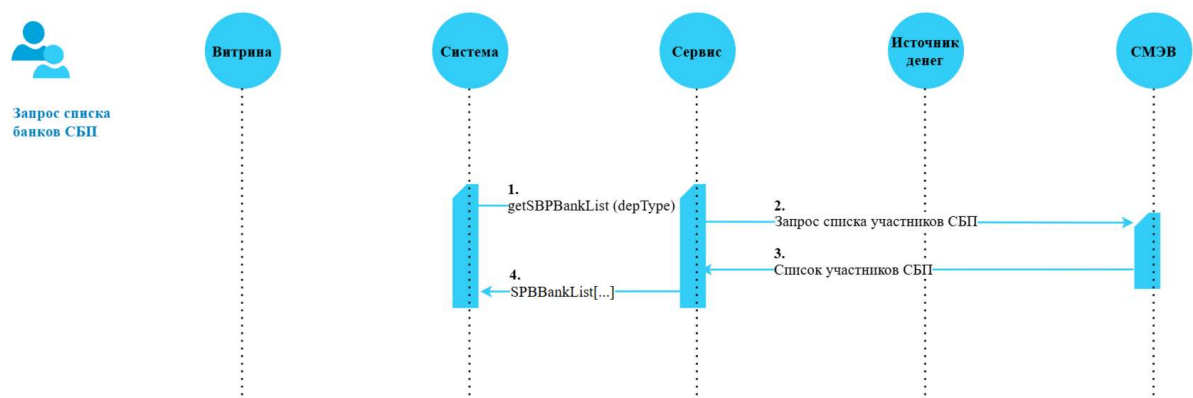


Рисунок 22. Схема загрузки списка банков СБП

## Параметры

### Входные параметры:

Название	Значение	Об.	Описание
depType	string	0..1	Определяет тип ведомства поиска (см. подробнее тип depType).

### **Выходные параметры:**

Параметр	Значение	Описание
SBPBankList	array	Список банков, подключенных к СБП.
memberId	string	Идентификатор банка, используется для заполнения targetAdditional в paymentParams.
memberName	string	Наименование Банка на английском языке.
memberType	string	Тип участника СБП.
memberNameRus	string	Наименование Банка на русском языке.

### Примеры

#### **Запрос:**

```
curl -i -X POST -H "Content-Type: application/json; charset=utf-8" -H
"Authorization: Bearer testservice" -d '{" depType ":"nft12345"}'
https://demopay.***.ru/api/multiserv/2.0/getSBPBankList
```

#### **Ответ:**

```
{ "SBPBankList":[
  {
    "memberId":"12345",
    "memberName":"Sber",
    "memberType":"_",
    "memberNameRus":"Сбер"
  },
  {
    "memberId":"12346",
    "memberName":"VTB",
    "memberType":"_",
    "memberNameRus":"ВТБ"
  },
  },
}
```

```

{
  "memberId": "12347",
  "memberName": "GPB",
  "memberType": "_",
  "memberNameRus": "Газпромбанк"
}
]
}

```

### 2.3.1.16 Метод checkSBPPayer

Запрос на проверку доступности переводов получателю по СБП для derType = sbpToSbp

Адрес запроса: POST https://demopay.\*\*\*.ru/api/multiserv/2.0/checkSBPPayer

#### Схема взаимодействия

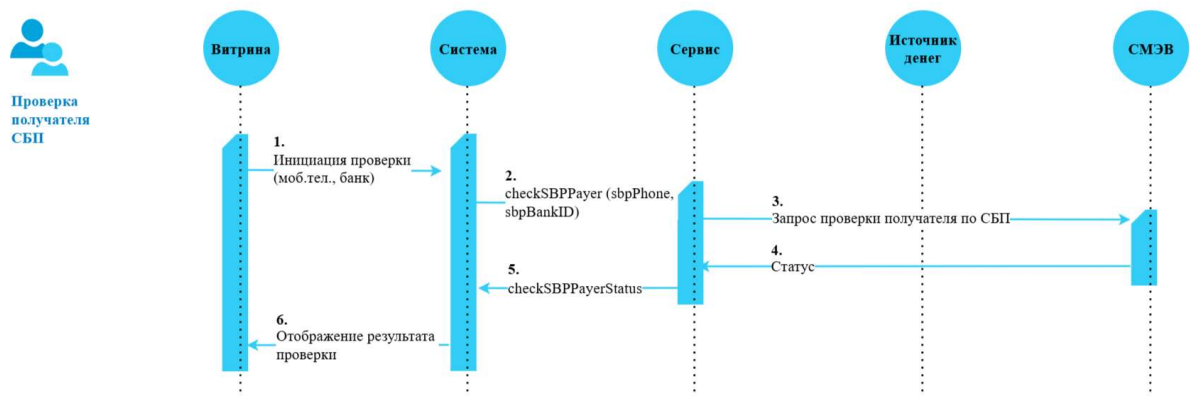


Рисунок 23. Схема проверки получателя СБП

#### Параметры

##### Входные параметры:

Название	Значение	Об.	Описание
paymentParams	object	1	Данные по оплачиваемому Начислению и Получателю платежа (см. подробнее тип paymentParams).

sbpPhone	string	1	Номер телефона Получателя (СБП), строка(20), (например, 79191013456).
sbpBankID	string	1	Уникальный идентификатор банка получателя (СБП), строка(12), пример 100000000124.

### **Выходные параметры:**

Параметр	Значение	Описание
checkSBPPayeeStatus	string	Статус проверки получателя: <ul style="list-style-type: none"> <li>• 1 – проверка успешно пройдена, переводы с СБП на СБП для данного получателя доступны;</li> <li>• 0 – проверка не пройдена, причины см. description.</li> </ul>
description	string	Описание причин отказа при проведении проверки получателя СБП.

### Примеры

#### **Запрос:**

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer testservice" -d '{
  "paymentParams": {
    "sbpPhone": "79191013456",
    "sbpBankID": "1crt88888881"
  }
}'
https://demopay.***.ru/api/multiserv/2.0/checkSBPPayee
```

#### **Ответ:**

```
{
  "checkSBPPayeeStatus": "0",
  "description": "Couldn't get information about receiver from SBP"
}
```

### 2.3.1.17 Метод refund

Полный или частичный возврат денежных средств Плательщику по платежу, доступен только для Партнеров являющихся непосредственными получателями денежных средств. Возврат осуществляется взаимозачетом. Если в течении трех рабочих дней в пользу получателя не будет достаточного объема платежей для компенсации суммы возврата, то будет выставлено требование на погашение задолженности. Доступен для платежей, проведенных через способ оплаты sbp. Операция возврат производится асинхронно, результат проведения возврата необходимо запросить вызовом метода paymentInfo указав в качестве идентификатора paymentID, также доступен сервис получения уведомлений о изменении статуса платежа

Адрес запроса: POST [https://demopay.\\*\\*\\*.ru/api/multiserv/2.0/refund](https://demopay.***.ru/api/multiserv/2.0/refund)

#### Схема взаимодействия

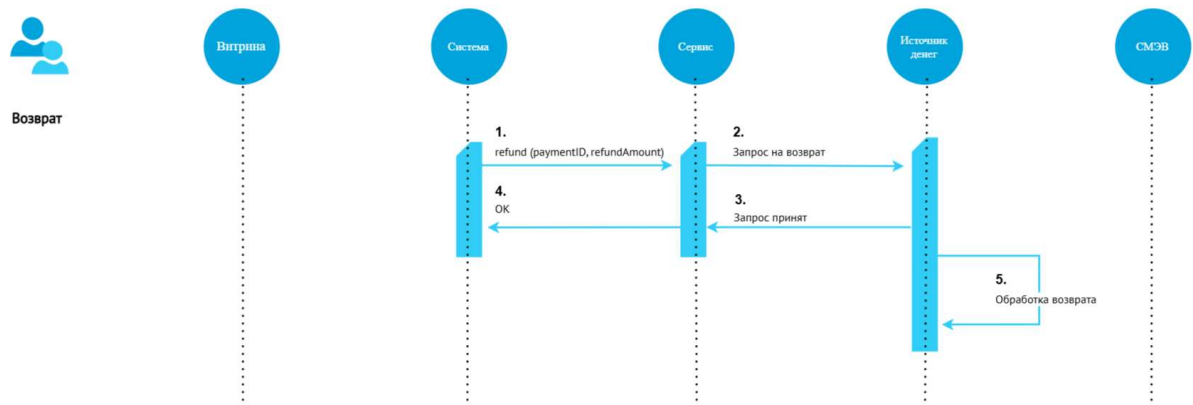


Рисунок 24. Схема полного или частичного возврата денежных средств

#### Параметры

##### Входные параметры:

Название	Значение	Об.	Описание
paymentID	string	да	Уникальный идентификатор платежа, по которому производится возврат, полученный в ответ на запрос pay по успешному платежу.

refundAmount	int	нет	Сумма возврата, в копейках, заполняется при частичном возврате, должна быть <= суммы платежа. Общая сумма возвратов по платежу должна контролироваться Системой, на стороне Сервиса ограничений нет.
sign	string	да	Подпись запроса. Должна быть сформирована как строка md5 от конкатенации значений параметров (paymentID, refundAmount, secret). Секретное слово индивидуально для каждого Партнера.

### Выходные параметры:

Параметр	Значение	Описание
status	string	Статус принятия запроса на возврат в обработку: <ul style="list-style-type: none"> <li>'0' =&gt; 'Запрос на возврат принят';</li> <li>'1' =&gt; 'Запрос на возврат в обработке';</li> <li>'2' =&gt; 'При запросе на возврат произошла ошибка'.</li> </ul>
message	string	Текстовая расшифровка статуса.

### **Примеры**

#### Запрос:

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer testservice" -d '{
  "paymentID": "6789",
  "refundAmount": 10000,
  "sign": "6ba0750aae540bd4a8510ea66df81453"
}'
https://demopay.***.ru/api/multiserv/2.0/refund
```

#### Ответ:

```
{
```

```

"status": "0",
"message": "Запрос на возврат принят"
}

```

### 2.3.1.18 Метод paymentInfoList

Получение информации по платежам за период, с возможностью запроса по статусу. Использование метода согласовывается с менеджером.

Адрес запроса: GET [https://demopay.\\*\\*\\*.ru/api/multiserv/2.2/paymentInfoList](https://demopay.***.ru/api/multiserv/2.2/paymentInfoList)

#### Схема взаимодействия

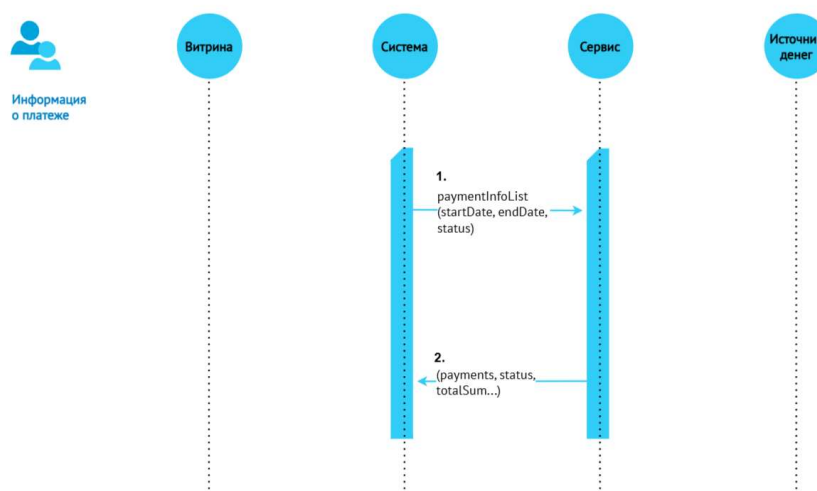


Рисунок 25. Схема получения информации о платежах по периоду

#### Параметры

Входные параметры:

Параметр	Тип	Об.	Описание
startDate	string	1	Дата начала периода отчета.
endDate	string	1	Дата окончания периода отчета.
status	string	0..1	Статус платежа, возможные значения: <ul style="list-style-type: none"> <li>created;</li> <li>auth;</li> </ul>

			<ul style="list-style-type: none"> <li>• cancel;</li> <li>• refunded.</li> </ul>
--	--	--	--

Выходные параметры:

Параметр	Тип	Описание
payments	array	Массив платежей, попавших в отчет.
paymentID	int	Идентификатор Платежа.
orderID	string	Уникальный номер Заказа на стороне Партнера.
status	object	Статус платежа в ПЦ (см. подробнее тип statuses).
code	int	Код статуса Платежа.
name	string	Наименование статуса Платежа.
date	string	Время установки статуса Платежа, указывается только для конечных статусов (auth и cancel).
cancelReason	string	Описание причины отмены платежа, указывается только для статуса cancel. (см. 4.9).
depType	string	Тип Платежа (см. подробнее тип depType).
kvit	boolean	Заказ Квитанции, возможные значения: <ul style="list-style-type: none"> <li>• true – с квитанцией;</li> <li>• false – без квитанции.</li> </ul>
amount	int	Сумма Платежа в пользу Получателя, в копейках.
payType	string	Способ оплаты, возможные значения: <ul style="list-style-type: none"> <li>• phone - со счета мобильного телефона;</li> <li>• card – банковской картой;</li> <li>• sbr – через сервис быстрых платежей ЦБ РФ.</li> </ul>
paymentParams	object	Параметры Платежа, переданные в запросе pay.

linkCard	boolean	Признак заказа привязки БК на витрине.
card	object	Параметры карты, с которой был совершен платеж.
pan	string	Маскированный номер карты. Отображаются первые 6 и последние 4 цифры.
card_type	string	Тип банковской карты (VISA, MASTERCARD, MIR и т.д.)
cardId	string	Идентификатор привязанной карты.

## Примеры

### Запрос:

```
curl -X POST -H "Content-Type: application/json; charset=utf-8" -H "Authorization: Bearer testservice" -d '{
  "startDate": "2024-12-02T10:00:00+03:00",
  "endDate": "2024-12-02T11:00:00+03:00"
}'
https://demopay.***.ru/api/multiserv/2.2/paymentInfoList
```

### Ответ (базовый пример):

```
{
  "payments": [
    {
      "paymentID": 21347,
      "orderID": "e10ccff0-504f-4cbb-9934-654fgty",
      "status": {
        "code": 20,
        "name": "auth",
        "date": "2024-12-02T10:03:25+03:00"
      },
      "depType": "merchant7842113520",
      "kvit": true,
    }
  ]
}
```

```
"amount" : 199900,
"totalSum" : 199900,
"payType" : "card",
"paymentParams":
  {
    "purpose": "Оплата подписки на 1 месяц"
  },
"card":
  {
    "pan":"427432*****5579",
    "card_type": " VISA"
  }
},
{
  "paymentID" : 21348,
  "orderID":"e20ccff0-504f-4cbb-9934-654fgty",
  "status" :
    {
      "code" : 30,
      "name" : "cancel",
      "date" : "2024-12-02T10:11:37+03:00",
      "cancelReason": "недостаточно средств"
    },
  "depType" : "merchant7842113520",
  "kvit": true,
  "amount" : 299900,
  "totalSum" : 299900,
  "payType" : "card",
  "paymentParams":
    {
      "purpose": "Оплата подписки на 1 месяц"
    },
  }
}
```

}

## 2.3.2 Описание типов данных

### 2.3.2.1 Ведомство (depType)

Определяет наименование ведомства, к которому относятся запрашиваемые Начисления.

Значение	Описание
gibdd	Штрафы и госпошлины Госавтоинспекции МВД России (ГИБДД).
fns	Налоги и сборы Федеральной налоговой службы (ФНС).
fssp	Задолженности Федеральной службы судебных приставов (ФССП).
rosreestr	Услуги Федеральной службы государственной регистрации, кадастра и картографии (Росреестр).
guvmmvd	Штрафы и госпошлины управления по вопросам миграции МВД (ГУ ВМ МВД).
rosgvard	Лицензии и штрафы Федеральной службы войск национальной гвардии (Росгвардия).
park**	Платные городские парковки, вместо ** необходимо указывать код региона.
parkFines**	Штрафы за платные городские парковки, вместо ** необходимо указывать код региона.
jkh	Услуги ЖКХ.
rnip	Госуслуги г. Москвы (транспорт, образование, недвижимость и т.д.)
donation	Пожертвования в пользу благотворительных, религиозных и общественных организаций.
justice	Судебные госпошлины.
freeFlow	Автодор, оплата задолженности за проезд по дорогам с системой взимания платы «Свободный поток».

sbpToSbp*	Переводы с СБП на СБП, вместо * указывается тарифная опция.
paidRoads	Оплата начислений из ГИС ГМП за проезд по платным дорогам.
unknown	Ведомство не определено.

### 2.3.2.2 Статусы Платежей (statuses)

Код	Наименование	Описание
1	created	Платеж создан
20	auth	Платеж авторизован
30	cancel	Платеж отменён, при использовании для оплаты витрины Сервиса в случае неуспеха предоставляется возможность повторить попытку.
40	refunded	Платеж возвращен, устанавливается из статуса auth

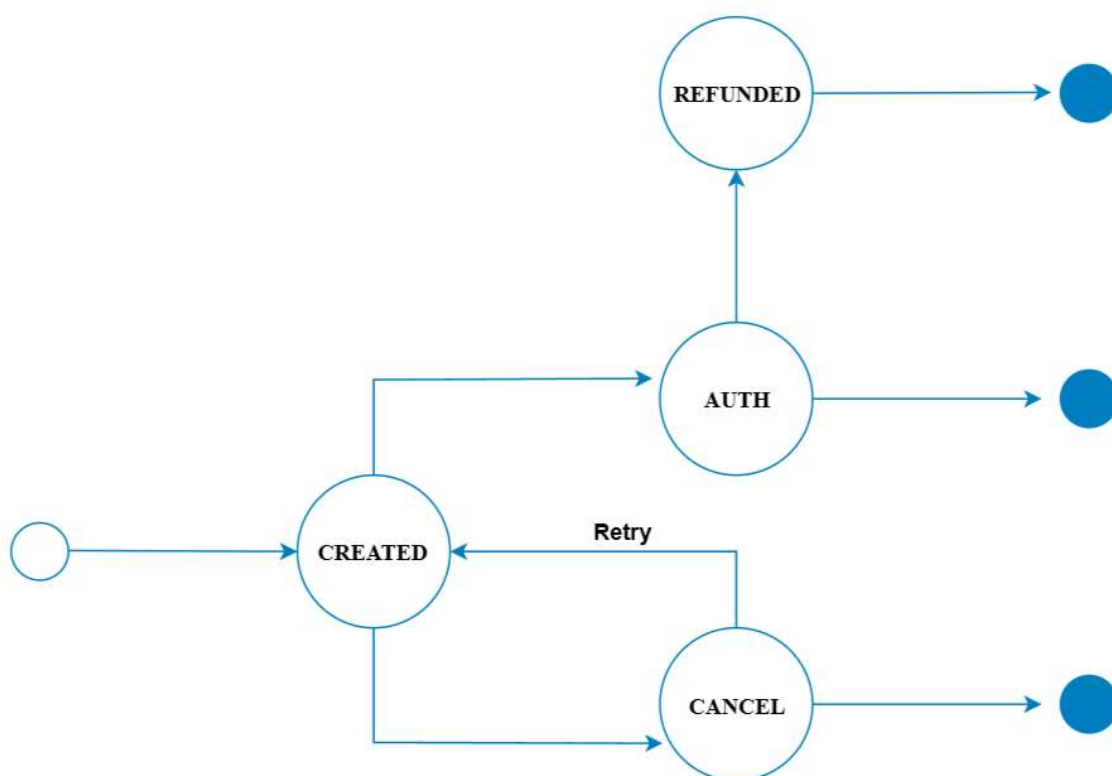


Рисунок 26. Схема изменений Статуса Платежа:

### 2.3.2.3 Идентификаторы Пользователя (payerDoc)

Определяют тип введенного Пользователем идентификатора документа Пользователя или исполнительного документа ведомства.

Описание структуры:

Параметр	Тип	Описание	Обязательность
code	string	Код идентификатора документа Пользователя или исполнительного документа ведомства	Да
value	array/ string	Массив документов Пользователя, для уведомления о новых начислениях возвращается одно значение.	Да
regionCode	int	Код региона Пользователя, 2 цифры	Нет
country	string	Страна, в которой зарегистрирован номер транспортного средства (ГРН) в формате Alpha-3 ISO 3166-1. Для РФ указывается RUS	Нет

Возможные значения:

Код документа	Описание	Формат
01	Паспорт РФ (используется для ФССП, ГУ ВМ МВД, Росреестр).	10 цифр.
02	Свидетельство о рождении (используется для Росреестр).	до 12 символов.
08	Паспорт иностранного гражданина.	до 12 символов.

14	СНИЛС (используется для ФССП, Росреестр, Росгвардия).	11 цифр.
15	ГРЗ (Государственный регистрационный номер транспортного средства).	Стандартные РФ: [АВЕКМНОРСТУХ]{1}\d{3} [АВЕКМНОРСТУХ]{2}\d{2,3}  Иные: [АВЕКМНОРСТУХДФГИЈЛНQRSUVWZ БГДЖЗИЙЛПФЦЧЩЪЫЬЭЮЯ0-9\-.]  <i>Например: RUSB8870A763, где в самом начале идет код страны.</i>
21	ИНН ФЛ (используется для ФССП, ФНС, ГУ ВМ МВД).	12 цифр, первые две не могут быть 00.
22	Водительское удостоверение (используется для ГИБДД).	10 цифр или 2 цифры + 2 буквы + 6 цифр, первые 2 цифры не могут быть 00.
24	Свидетельство о регистрации ТС (используется для ГИБДД).	10 цифр или 2 цифры + 2 буквы + 6 цифр, первые 2 цифры не могут быть 00.
51	Номер исполнительного или административного производства ФССП.	0000000/00/0000-ИП 0000000/00/00/00 0000000/00/0000-АП
91	Единый лицевой счет (ЕЛС). Лицевой счет, автоматически формируемый ГИС ЖКХ, и включающий информацию о помещении и о лице, обязанном вносить плату за жилое	11ВА456331 2 цифры + 2 буквы+ 6 цифр.

	помещение и коммунальные услуги.	
92	Идентификатор жилищно-коммунальных услуг (ИЖКУ).	11BA456331-03 2 цифры + 2 буквы+ 6 цифр + знак "-" + 2 цифры.
93	Идентификатору платежного документа (ИПД). Номер платежного документа, включающий идентификатор жилищно-коммунальной услуги и информацию о платежном документе.	11BA456331-03-7223 2 цифры + 2 буквы+ 6 цифр + знак "-" + 2 цифры + знак "002D" + 4 цифры.
99	Уникальный идентификатор Начисления:  Номер постановления об административном правонарушении ГИБДД;  Индекс налогового документа ФНС;  Код оплаты услуги Росреестра;  Начисление федерального или муниципального органа власти.	20 цифр или 25 цифр.
931	ФИО (все буквы прописные) и Дата рождения (формат даты ГГГГММДД). Разделитель пробел. Для данного типа	ИВАНОВ РОМАН ИВАНОВИЧ 19710318

	документа обязательно заполнение параметра regionCode.	
200	Идентификатор юридического лица резидента РФ.	ИНН (10 цифр) и КПП (9 симв.) 7717751677771801001
300	Идентификатор юридического лица нерезидента РФ.	ИНН (10 цифр) + КПП (9 симв.) 7717751677771801001 00000 + КИО (5 симв.) + КПП (9 симв.) 0000051677771801001 Код налогоплательщика-юридического лица в стране регистрации без «№», «-» и др. разделителей (14 симв, дополняется слева 0) + буквенный код страны по ОКСМ (2 симв.) + цифровой код страны по ОКСМ (3 цифр.) 00000123456789DE276
400	Идентификатор индивидуального предпринимателя.	0000000 + ИНН (12 цифр) 0000000226606001503

#### 2.3.2.4 Коды ответов при возникновении ошибки при обработке запроса (error)

##### Описание структуры

Параметр	Тип	Описание
code	int	Код ошибки
message	string	Детальное описание ошибки
details	string	Дополнительная информация (необязательное)

##### Возможные значения

Определяют результат обработки запроса:

Код	Описание ошибки
100	Общая ошибка. В результате выполнения работы критическая ошибка.
101	Ошибка валидации входящего запроса.
102	Ошибка, отсутствует обязательный параметр.
103	Невалидный токен запроса.
104	Невалидная подпись под запросом.
105	Не найдены данные по переданному для оплаты начислению.
106	Платёж по данным параметрам уже существует.
107	Платеж не найден.
108	Квитанция предоставляется только по авторизованным платежам.
109	Баланс счета обеспечения недостаточен для проведения платежа.
110	Ошибка при привязке банковской карты.
111	Ошибка при удалении привязанной банковской карты.
112	Отсутствуют права на предоставлении информации по платежу.
113	По указанному платежу отсутствует квитанция.
114	Неправильно указан параметр PayId.
115	Для данного Пользователя авторизация временно заблокирована.
116	Введен не верный код 3-D Secure.
117	Банковская карта уже привязана.
118	Не используется payerID, необходимо передавать значение параметра phone.
119	Не используется phone, необходимо передавать значение параметра payerID.
120	Параметр totalSum рассчитан неверно.

121	Неподдерживаемый способ оплаты.
122	Неверно сформирован JSON.
123	Некорректное название параметра <параметры с ошибкой в названии>.
124	Платеж по данным реквизитам невозможен, не найден УИН: <список не найденных УИН, разделитель запятая>.
125	Платеж, по которому происходит привязка - не завершен или не успешен.
126	Транзакция, по которой происходит привязка не прошла 3-D Secure авторизацию.
127	Неправильно указан входной параметр <невалидный параметр>.
128	Отсутствуют права на предоставлении платежного поручения.
129	Платежное поручение ещё не выгружено. Повторите запрос позднее.
130	Запрос платежных поручений недоступен. Обратитесь к менеджеру.
131	Платежное поручение предоставляется только по авторизованным платежам.
132	Сервис запроса ИНН физического лица по паспортным данным временно недоступен, повторите запрос позднее.
133	Общая сумма платежей в пакете (packetAdvancePayments) не совпадает с amount.
134	Превышен лимит количества запросов к API, повторите запрос позднее.
135	Сервис генерации уведомления о платеже временно недоступен, повторите запрос позднее.
136	Количество платежей при мультиплате не может быть более 500 в одном запросе.
137	Код платежа Росреестра – получатель не найден.
138	Код платежа Росреестра – услуга не найдена.
139	Код платежа Росреестра – заявка просрочена.

140	Код платежа Росреестра – неверная контрольная сумма.
141	Код платежа Росреестра – неверный тип заявки.
142	Неверный тип для авансовой оплаты.
143	Общая сумма платежей в мультиплатеже не совпадает с amount.
144	Сумма платежа меньше amount.
145	В операции по данным платежным реквизитам отказано.
146	Подписка не найдена.
147	Ошибка при получении суммы задолженностей за проезды по платным дорогам с системой взимания платы "Свободный поток" [текст ошибки].
154	Запрос к BarcodeApi завершился с ошибкой: Ошибка (50). Response Message: фатальная техническая ошибка.
155	Подписка отменена.
156	Не удалось найти подписку по subscriptionUuid.
157	Оплата с totalSum не разрешена.
158	Неверно рассчитан totalSum.
159	Сумма Контракта меньше или равна нулю.

### 2.3.2.5 Данные по найденному Начислению (charge)

#### **Параметры найденного Начисления:**

Название	Значение	Описание
payerDoc	array	Идентификатор Пользователя, по которому найдено начисление (см. подробнее тип <u>payerDoc</u> ).
unifiedPayerIdentifier	string	Идентификатор юридического лица. Заполняется только в случае начисления на юридическое лицо.

depType	string	тип ведомства (см. подробнее тип depType).
supplierBillID	string	Уникальный идентификатор Начисления, для налогов оплачиваемых на ЕНС = 0.
mainSupplierBillID	array	Идентификаторы начислений, на основании которых выставлено данное Начисление.
paymentDocumentID	string	Идентификатор платежного документа.
chargeData	object	Реквизиты найденного Начисления.
billDate	string	Дата выставления Начисления.
validUntil	string	Дата, до которой (включительно) актуально выставленное Начисление.
amountToPay	int	Сумма к оплате в выставленном Начислении, указывается в копейках.
accountNumberJkh	string	Лицевой счет Пользователя услуг ЖКХ.
addressUserJkh	object	Адрес пользователя для оплаты ЖКХ (см. подробнее тип addressUserJkh).
purpose	string	Назначение Платежа.
kbk	string	КБК.
oktmo	string	ОКТМО.
inn	string	ИНН Получателя.
kpp	string	КПП Получателя.
bik	string	БИК банка Получателя.
correspondentBankAccount	string	Корреспондентский счет банка Получателя.
accountNumber	string	Счет Получателя.

payeeName	string	Наименование Получателя.
divID	int	Идентификатор Получателя (опциональное поле).
amount	int	Сумма начисления, указывается в копейках.
payPeriod	object	Оплачиваемый период (для платежей ЖКХ).
year	string	Год.
month	string	Месяц.
payerIdentifier	string	Идентификатор Пользователя.
additionalPayerIdentifier	string	Дополнительный идентификатор Пользователя.
payerName	string	Наименование Плательщика по начислению (ФИО).
treasureBranch	string	ТОФК (территориальный орган федерального казначейства).
additionalData	object	Дополнительные параметры Начисления (инф. о скидке и статье КоАП для штрафа ГИБДД, дополнительные параметры ИП ФССП и т.д.)
discountSize	string	Размер скидки.
discountDate	string	Срок действия скидки.
spiNamePhone	string	ФИО и номер телефона судебного пристава исполнителя.
dataIpNumber	string	Идентификатор исполнительного производства.
ipDebtRestIp	string	Остаток задолженности по исполнительному производству.

ID	string	Основание для возбуждения ИП.
IP	string	Исполнительное производство (дата и номер).
IDOrgName	string	Наименование инициатора ИП.
legalAct	string	Статья КоАП.
offenseName	string	Наименование правонарушения.
offensePlace	string	Место правонарушения.
carNameSRM	string	Транспортное средство.
offenseDate	string	Дата правонарушения.
offenceCoordinates	string	Координаты нарушения.
fixDateTime	string	Время фиксации нарушения.
gracePeriod	string	Грейс период.
TRX_DHMS	string	Дата и время проезда.
trip	string	Маршрут.
direction	string	Пункт взимания платы.
vehicleClass	string	Класс ТС.
changeStatus	object	Статус изменения начисления.
plazaTransactions	array	Список идентификаторов поездок.
paymentData	object	Информация по успешным платежам, проведенным в пользу данного начисления.
sum	int	Сумма всех успешных Платежей, проведенных в пользу данного Начисления, указывается в копейках.

date	string	Дата оплаты последнего успешного Платежа в пользу данного Начисления.
ttl	string	время актуальности Начисления.

### 2.3.2.6 Информация о Пользователе (payerParams)

#### **Информация о Пользователе, необходимая для проведения Платежа:**

Название	Значение	Об.	Описание
phone	int	0..1	Номер телефона Пользователя, 10 цифр, (необязательный в случае оплаты методом card, deposit и др.), должен быть заполнен для paidRoads, если не заполнен email.
fio	string	0..1	ФИО Пользователя (необязательный для платежей отправляемы сводным п.п. по договору).
email	string	0..1	Адрес эл. почты Пользователя (необязательный), должен быть заполнен для paidRoads, если не заполнен phone.
address	string	0..1	Адрес регистрации Пользователя (необязательный).
payerDoc	array	0..1	Идентификатор, введенный Пользователем для идентификации (см. подробнее тип payerDoc), могут быть использованы (паспорт РФ, ИНН, СНИЛС, номер ВУ, СТС). Для авансовых бюджетных платежей обязательный параметр (один идентификатор). При наличии УИН !=0 или для не бюджетных платежей (ЖКХ) не заполняется. При УИН =0 и depType=fns обязательно заполнен значениями ИНН ФЛ/ИП (21, 400) или идентификатором ЮЛ (200,300).

unifiedAccountNumber	string	0..1	Единый лицевой счет Пользователя в ГИС ЖКХ.
accountNumber	string	0..1	Лицевой счет Пользователя услуг ЖКХ.
additionalData	object	0..1	Дополнительные параметры Пользователя.

### 2.3.2.7 Данные по Получателю платежа (paymentParams)

Данные по получателю платежа:

Название	Значение	Об.	Описание
supplierBillID	array/string	0..1	Уникальный идентификатор Начисления, для Авансовых платежей не указывается: <ul style="list-style-type: none"> <li>string - для платежа с одним начислением;</li> <li>array - для мультиплатежа по УИН.</li> </ul>
paymentDocumentID	array/string	0..1	Уникальный идентификатор платежного документа ГИС ЖКХ: <ul style="list-style-type: none"> <li>string - для платежа с одним начислением;</li> <li>array - для мультиплатежа по ИПД.</li> </ul>
packetAdvancePayments	array	0..1	Пакет с реквизитами получателей для мультиоплаты авансовых платежей см. тип packetAdvancePayments.
purpose	string	0..1	Назначение Платежа.
kbk	string	0..1	КБК.
oktmo	string	0..1	ОКТМО.
inn	string	0..1	ИНН Получателя.

kpp	string	0..1	КПП Получателя.
bik	string	0..1	БИК банка Получателя.
correspondentBankAccount	string	0..1	Корреспондентский счет банка Получателя.
accountNumber	string	0..1	Счет Получателя.
payeeName	string	0..1	Наименование Получателя.
treasureBranch	string	0..1	ТОФК.
ls	string	0..1	Лицевой счет Получателя.
accountNumberJkh	string	0..1	Лицевой счет Пользователя услуг ЖКХ.
payPeriod	object	0..1	Оплачиваемый период (для платежей ЖКХ).
year	string	1	Год.
month	string	1	Месяц.
targetAccount	string	0..1	Номер мобильного телефона получателя, по которому будет выполнен перевод через СБП, 10 цифр, первая 9, обязателен для перевода на СБП.
targetAdditional	string	0..1	Идентификатор Банка получателя в СБП и регион оплаты. Список параметров в формате: MemberID=nnn Reason=nn <ul style="list-style-type: none"> <li>MemberID – идентификатор банка получателя в СБП. Список всех банков получатлей по СБП до вызова pay необходимо запросить через метод getSBPBankList;</li> </ul>

			<ul style="list-style-type: none"> <li>Reason (не обязательное) – код территории, где совершена операция, по ОКТМО (2 цифровых знака).</li> </ul> <p>Обязателен для перевода на СБП.</p> <p><i>Например:</i>  <i>MemberID=123 Reason=12</i></p>
country	string	0..1	Страна, в которой зарегистрировано номер транспортного средства (ГРН) в формате Alpha-3 ISO 3166-1, обязателен для depType=freeFlow
grnz	string	0..1	Государственный регистрационный номер транспортного средства (ГРН) обязателен для depType=freeFlow.
plazaTransactions	array	0..1	Список идентификаторов поездок.
sbpPhone	string	0..1	Номер телефона Получателя (СБП), строка(20), пример 79191013456.
sbpBankID	string	0..1	Уникальный идентификатор банка получателя (СБП), строка(12), пример 100000000124.
sbpDescription	string	0..1	Описание перевода (СБП).
additionalData	object	0..1	Дополнительные параметры Платежа.

### 2.3.2.8 Пакет с реквизитами получателей для мультиплаты авансовых платежей (packetAdvancePayments)

Данные по получателю платежа:

Название	Значение	Об.	Описание
supplierBillID	string	0..1	Уникальный идентификатор Начисления, в случае отсутствия заполняется 0.

paymentDocumentID	array/string	0..1	Уникальный идентификатор платежного документа ГИС ЖКХ.
amount	int	1	Сумма платежа без учета комиссий, указывается в копейках.
totalSum	int	0..1	Сумма платежа с учетом комиссий, которая будет списана с Пользователя, в копейках, используется при расчете суммы списания с Пользователя ИС Партнера, использование параметра возможно только после согласования с менеджером. В случае использования с мультиплатежами должна быть указана для каждой операции.
purpose	string	1	Назначение Платежа.
kbk	string	0..1	КБК.
oktmo	string	0..1	ОКТМО.
inn	string	1	ИНН Получателя.
kpp	string	1	КПП Получателя.
bik	string	1	БИК банка Получателя.
correspondentBankAccount	string	0..1	Корреспондентский счет банка Получателя.
accountNumber	string	1	Счет Получателя.
payeeName	string	1	Наименование Получателя.
ls	string	0..1	Лицевой счет Получателя.
accountNumberJkh	string	0..1	Лицевой счет Пользователя услуг ЖКХ.

### 2.3.2.9 Адрес пользователя для оплаты услуг ЖКХ (addressUserJkh)

Название	Значение	Описание
Region	string	Регион.
City	string	Город.
Street	string	Улица.
HouseNum	string	Номер дома.
FIASHouseGuid	string	Глобальный уникальный идентификатор дома по ФИАС.
Apartment	string	Номер квартиры, апартаментов.
AddressString	string	Полный адрес.

### 2.3.2.10 Причины отмены платежа (cancelReason)

Возможные значения:

Описание причины
Платеж отменен, т.к. Клиент не авторизовал его в течение одного (1) часа после создания.
Платеж явно отменен Клиентом.
Платеж отменен ЭМом.
Платеж отменен тех. поддержкой Компании.
При обработке Платежа возник внутренний сбой в Сервисе.
Клиент ошибся критическое число раз при вводе данных.
В обработке операции по указанным платежным реквизитам отказано.
Операции по указанным платежным реквизитам временно запрещены.
Недостаточно средств.

Неверный код CVC2/CVV2.
Карта просрочена.
Отклонено платежной системой.
Техническая ошибка взаимодействия с платежной системой.
Карта не найдена.
Ошибка 3-D Secure.
Превышен лимит суммы операций по карте.
Превышен лимит активности использования карты.

### 2.3.2.11 Криптограмма

Возможность проведения платежей через криптограмму согласовывается с менеджером. Для формирования криптограммы должен использоваться скрипт с сайта Сервиса, ссылка на скрипт выдается Партнеру при подключении к Сервису. Скрипт размещается на витрине Партнера в целях формирования криптограммы алгоритмом RSA.

Криптограмма передается по защищенному каналу (SSL) на сервер Партнера, затем передается в Сервис, в запросе `pay`. Взаимодействие витрины Партнера с Сервисом напрямую запрещено.

#### Требования к форме ввода данных:

- У полей с карточными данными не должно быть атрибута "name", чтобы данные не попали на сервер при отправке формы.
- Данные из формы должны передаваться по SSL.

#### Требования к криптограмме:

- Должна формироваться оригинальным скриптом, загруженным с сайта Сервиса
- Криптограмму нельзя сохранять и использовать повторно

#### Требования к безопасности по PCI DSS:

С точки зрения PCI DSS, подобный способ подключения классифицируется как "E-commerce merchants who outsource all payment processing to PCI DSS validated third parties, and who have a website(s) that doesn't directly receive cardholder data but

that can impact the security of the payment transaction. No electronic storage, processing, or transmission of any cardholder data on the merchant's systems or premises.", то есть обработка платежных данных выполняется третьей стороной, но сайт влияет на безопасность карточных данных.

Для соблюдения требований стандарта, необходимо заполнять лист самооценки SAQ-EP и ежеквартально проходить ASV тестирование. Сканирование должно проводиться аккредитованным вендором (ASV) из списка, представленного на сайте совета PCI.

## Примеры

Поля с данными карты должны быть помечены следующими атрибутами:

- data-mm="cardNumber" - поле с номером карты
- data-mm="expDateMonthYear" - поле со сроком действия в формате ММYY
- data-mm="expDateMonth" - поле с месяцем срока действия
- data-mm="expDateYear" - поле с годом срока действия
- data-mm="cvv" - поле с кодом CVV
- data-mm="name" - поле с именем держателя карты

### 2.3.2.12 Подпись (sign)

#### Пример формирования подписи для рекуррентного платежа

```
{
  "orderID": "12345005",
  "payerID": "abc098xyz",
  "depType": "fns",
  "payType": "sbp",
  "kvit":true,
  "sbpSubscription": {
    "subscriptionUuid": "2b265fe1-aad8-469a-92dc-4fdb7abab7c0"
  },
  "payerParams": {
    "fio": "Иванов Иван Иванович",
    "email": "test@email.com"
  },
  "paymentParams": {
    "supplierBillID": "18209966253930126189"
  },
  "sign":"{{hash}}"
}
```

Строка sign должна быть сформирована как строка md5 от конкатенации значений параметров (kvit + supplierBillID + payerID + subscriptionUuid + secretKey ).

Для объектов в конкатенации участвуют значения всех параметров внутри него. Значение параметра kvit должно быть приведено к int (true – 1, false – 0).

Секретное слово индивидуально для каждого Партнера.

В примере:

```
118209966253930126189abc098xyz2b265fe1-aad8-469a-92dc-4fdb7abab7c0CRYPTO
```

в итоге получаемое значение:

```
25a2db21db014b798185f437f512d185
```

### Пример формирования подписи для рекуррентного платежа банковской картой

```
{
  "orderID": "12345",
  "payerID": "user12345",
  "cardID": 123,
  "depType": "gibdd",
  "kvit": true,
  "payerParams": {
    "fio": "Иванов Иван Иванович"
  },
  "paymentParams": {
    "supplierBillID": "18810177170712879661"
  },
  "sign": "{{hash}}"
}
```

Строка sign должна быть сформирована как строка md5 от конкатенации значений параметров (kvit + supplierBillID + payerID + cardID + secret key). Для объектов в конкатенации участвуют значения всех параметров внутри него. Значение параметра kvit должно быть приведено к int (true – 1, false – 0).

Секретное слово индивидуально для каждого Партнера.

В примере:

```
118810177170712879661user12345123CRYPTO
```

в итоге, получаемое значение:

```
133742e20a8ac8901853df88978772c9
```

### Пример формирования подписи для обычного платежа банковской картой

```
{
  "orderID": "12345",
  "depType": "paidRoads",
  "payType": "card",
}
```

```

"kvit":true,
"payerParams":{
  "fio":"Иванов Иван Иванович",
  "email":"ivanov@mail.ru"
},
"paymentParams":{
  "supplierBillID":"0412479001000013586054242"
},
"returnUrl":"https://testpartner.ru/e0462fc0872e2/",
"sign":"{{hash}}"
}

```

Строка sign должна быть сформирована как строка md5 от конкатенации значений параметров (payType + kvit + amount + supplierBillID + secretKey). Для объектов в конкатенации участвуют значения всех параметров внутри него. Значение параметра kvit должно быть приведено к int (true – 1, false – 0).

Секретное слово индивидуально для каждого Партнера.

В примере:

```
card10412479001000013586054242CRYPTO
```

в итоге, получаемое значение:

```
133742e20a8ac8901853df88978772c9
```

### Пример формирования подписи для авансового платежа

```

{
  "orderID":"1234567891",
  "depType":"gibdd",
  "payType":"card",
  "kvit":false,
  "amount":10000,
  "payerParams":{
    "phone":"9111234574",
    "fio":"Иванов Иван Иванович",
    "address":"Школьная 25",
    "email":"test@test.com",
    "payerDoc":[
      {
        "code":"22",
        "value":[
          "1234567890"
        ]
      }
    ]
  },
  "paymentParams":{

```

```

    "purpose": "УИН18810067170000804114//ШТРАФ ПО АДМИНИСТРАТИВНОМУ
ПРАВОНАРУШЕНИЮ, № 18810067170000804114 от 17.07.2017",
    "kbk": "18210807010011000110",
    "oktmo": "45367000",
    "inn": "7733506810",
    "kpp": "773301001",
    "bik": "048327001",
    "accountNumber": "40101810800000010041",
    "payeeName": "УФК по г. Москве (МИФНС России № 46 по г. Москве)"
  },
  "returnUrl": "http://testpartner.ru/e0462fc0872e9/",
  "sign": "f1e4803a872c6a18299520c617063ec3"
}

```

Строка sign должна быть сформирована как строка md5 от конкатенации значений параметров (payType + kvit + amount + paymentParams + secretKey). Для объектов в конкатенации участвуют значения всех параметров внутри него. Значение параметра kvit должно быть приведено к int (true – 1, false – 0).

Секретное слово индивидуально для каждого Партнера.

В примере:

```

card010000УИН18810067170000804114///ШТРАФ ПО
АДМИНИСТРАТИВНОМУ ПРАВОНАРУШЕНИЮ, № 18810067170000804114 от
17.07.2017182108070100110001104536700077335068107733010010483
2700140101810800000010041УФК по г. Москве (МИФНС России № 46
по г. Москве) CRYPTO

```

в итоге, получаемое значение:

```
2136d782a19d86d1b11e85135a570e5c
```

### Пример формирования подписи для платежа с paymentParams:

```

// получить параметры из тела запроса в формате json
var requestData = JSON.parse(request.data);
// обязательные параметры
var kvit = requestData.kvit ? 1 : 0;
// не обязательные параметры
var defPayType = requestData.defPayType;
var payType = requestData.payType;
var amount = "";
if (request.data.has("amount")) {
    amount = requestData.amount;
}
var pmToSign = collectValuesFromObj(requestData.paymentParams)
function collectValuesFromObj(obj) {
    var collectValues = '';

```

```

    for (var k in obj) {
        if (typeof obj[k] === 'object') {
            collectValues += collectValuesFromObj(obj[k]);
        } else if (typeof obj[k] === 'string') {
            collectValues += obj[k];
        }
    }
    return collectValues;
}
var secretKey = "secret";

// CryptoJS
var hash = CryptoJS.MD5(kvit + pmToSign + secretKey).toString();

// Объявить новую переменную окружения и установить ей значение
postman.setEnvironmentVariable('hash', hash);

```

Для авансового платежа с packetAdvancePayments и defPayType:

```

// получить параметры из тела запроса в формате json
var requestData = JSON.parse(request.data);
// обязательные параметры
var payType = requestData.payType;

if (requestData.has("defPayType")) {
    payType = '';
}
var kvit = requestData.kvit ? 1 : 0;
// не обязательные параметры
var amount = "";
if (requestData.has("amount")) {
    amount = requestData.amount;
}
var advancePaymentData = '';
var advancePayments = requestData.paymentParams.packetAdvancePayments;
advancePayments.forEach(function(arr) {
    // Проходимся по каждому полю в элементе массива
    for (var element in arr) {
        // Проверяем наличие свойства у массива
        if (arr.hasOwnProperty(element)) {
            // Приводим все к строкам чтобы сложить поля как строки, а не как числа
            advancePaymentData += String(arr[element]);
        }
    }
});
var secretKey = "secret";

//

```

```
var hash = CryptoJS.MD5(payType + kvit + amount + advancePaymentData +
secretKey).toString();

// Объявить новую переменную окружения и установить ей значение
postman.setEnvironmentVariable('hash', hash);
```

## 2.3.3 Дополнительная информация по структурам и сценариям работы

### 2.3.3.1 Структура начислений и уведомлений

Массив уведомлений по новым/измененным/оплаченным/аннулированным Начислениям, найденным по Идентификаторам Пользователя. Для активации подписки Партнеру необходимо предоставить менеджеру список типов уведомлений, которые планируется получать и url для отправки Сервисом уведомлений.

Для внесения в список вайтлист Партнеру необходимо запросить у менеджер список IP адресов, с которых осуществляется отправка уведомлений.

Схема отправки уведомления, в случае неуспешной доставки:

- сразу после наступления события;
- через 30 секунд после предыдущей попытки;
- через 1 минуту после предыдущей попытки;
- через 5 минут после предыдущей попытки;
- через 20 минут после предыдущей попытки;
- через 60 минут после предыдущей попытки.

### Параметры

Название	Значение	Об.	Описание
subscribe	string	1	Уникальный идентификатор Подписки Пользователя.
msgType	int	1	Тип уведомления, возможные значения: <ul style="list-style-type: none"><li>• 1 – новое начисление;</li><li>• 2 – уведомление об изменении начисления;</li><li>• 3 – начисление аннулировано;</li><li>• 4 – начисление оплачено.</li></ul>
payerDoc	object	1	Идентификатор Пользователя, по которому найдено начисление (см. подробнее тип <code>payerDoc</code> ).
depType	string	1	Тип ведомства (см. подробнее тип <code>depType</code> ).

supplierBillID	string	1	Уникальный идентификатор Начисления.
billDate	string	1	Дата выставления Начисления.
payerIdentifier	string	0..1	Идентификатор Получателя Платежа.
additionalPayerIdentifier	string	0..1	Дополнительный идентификатор Получателя Платежа.
amountToPay	int	0..1	Сумма к оплате в выставленном Начислении, указывается в копейках.
purpose	string	1	Назначение Платежа.
payeeInn	string	0..1	ИНН Получателя Платежа.
payeeKpp	string	0..1	КПП Получателя Платежа.
payeeAccount	string	0..1	Счет Получателя Платежа.
payeeBik	string	0..1	БИК Получателя Платежа.
payeeCorrAccount	string	0..1	Корреспондентский счет Получателя Платежа.
payeeBankName	string	0..1	Банк Получателя Платежа.
kbk	string	1	КБК.
payeeName	string	1	Наименование Получателя.
oktmo	string	0..1	ОКТМО.
amount	int	1	Сумма начисления, указывается в копейках.
payerName	string	0..1	Наименование Плательщика по начислению (ФИО).
additionalDataDiscountSize	string	0..1	Размер скидки, в %.

additionalDataDiscountDate	string	0..1	Срок действия скидки.
departmentName	string	0..1	Ведомство.
legalAct	string	0..1	Статья КоАП.
offenseDate	string	0..1	Дата правонарушения.
offensePlace	string	0..1	Место правонарушения.

## Примеры структуры

### ГИБДД

```

{
  "subscribe ": "53e0969e-efcb-43a8-a06e-3c1fc516cbca ",
  "payerDoc ": {
    "code ": "24 ",
    "value ": "7604644353 "
  },
  "depType ": "gibdd ",
  "supplierBillID ": "18810536250512168108 ",
  "billDate ": "2025-05-12T00:00:00+03:00 ",
  "amountToPay ": 75000,
  "purpose ": "Оплата штрафа по постановлению 18810536250512168108 от
12.05.2025 ",
  "kbk ": "18811601121010001140 ",
  "payeeName ": "УФК ПО ВОРОНЕЖСКОЙ ОБЛАСТИ (ГУ МВД России по
Воронежской обл.) ",
  "amount ": 75000,
  "payerName ": "- ",
  "additionalDataDiscountSize ": "25 ",
  "additionalDataDiscountDate ": "2025-06-11 ",
  "msgType ": 1
}

```

### Ространснадзор

```
{
  "subscribe ": "2400000000007415029903643 ",
  "payerDoc ": {
    "code ": "24 ",
    "value ": "7415029903 "
  },
  "depType ": "gibdd ",
  "supplierBillID ": "10673342253463593852 ",
  "billDate ": "2025-04-09T00:00:00+03:00 ",
  "amountToPay ": 0,
  "purpose ": "Штраф за неоплату возмещения вреда дорогам федерального значения по статье 12.21.3 КоАП ч.1 и ч.2 ",
  "kbk ": "10611601121010005140 ",
  "payeeName ": "УФК по г. Москве (МТУ Ространснадзора по ЦФО л/с №04731БФ3540) ",
  "amount ": 500000,
  "payerName ": "БРЕХОВ АНДРЕЙ ГЕННАДЬЕВИЧ ",
  "additionalDataDiscountSize ": "25 ",
  "additionalDataDiscountDate ": "2025-05-12 ",
  "msgType ": 4
}
```

### Парковки

```
{
  "subscribe ": "sts-9930736983 ",
  "payerDoc ": {
    "code ": "24 ",
    "value ": "9930736983 "
  },
  "depType ": "gibdd ",
  "supplierBillID ": "0318948825022788000002339 ",
  "billDate ": "2025-03-03T00:00:00+03:00 ",
  "amountToPay ": 0,
```

```
"purpose ": "Пост. №85100001250227002260 ст. 3.16Фото: чат-бот
KZN_ADMKOM в Telegram или https://kzn.ru/personal/fine/ пройдя авторизацию в Личн.
кабинете через Госуслуги. Тел.:+7(843)223-23-96, admkom.kazan@tatar.ru ",
"kbk ": "85111602020020000140 ",
"payeeName ": "МКУ «Управление административно-технической инспекции
Исполнительного комитета муниципального образования города Казани» ",
"amount ": 250000,
"payerName ": "Мусин Марсель Ильясович ",
"additionalDataDiscountSize ": " ",
"additionalDataDiscountDate ": " ",
"msgType ": 4
}
```

### Мособл

```
{
"subscribe ": "sts-9930736983 ",
"payerDoc ": {
"code ": "24 ",
"value ": "9930736983 "
},
"depType ": "gibdd ",
"supplierBillID ": "0318948825022788000002339 ",
"billDate ": "2025-03-03T00:00:00+03:00 ",
"amountToPay ": 0,
"purpose ": "Пост. №85100001250227002260 ст. 3.16Фото: чат-бот
KZN_ADMKOM в Telegram или https://kzn.ru/personal/fine/ пройдя авторизацию в Личн.
кабинете через Госуслуги. Тел.:+7(843)223-23-96, admkom.kazan@tatar.ru ",
"kbk ": "85111602020020000140 ",
"payeeName ": "МКУ «Управление административно-технической инспекции
Исполнительного комитета муниципального образования города Казани» ",
"amount ": 250000,
"payerName ": "Мусин Марсель Ильясович ",
"additionalDataDiscountSize ": " ",
"additionalDataDiscountDate ": " ",
"msgType ": 4
}
```

```
}
```

### АМПП

```
{  
  "subscribe ": "2400000000009917109145643 ",  
  "payerDoc ": {  
    "code ": "24 ",  
    "value ": "9917109145 "  
  },  
  "depType ": "gibdd ",  
  "supplierBillID ": "035543100000000060632118 ",  
  "billDate ": "2025-05-12T00:00:00+03:00 ",  
  "amountToPay ": 0,  
  "purpose ": "Оплата за перемещение ТС на спецстоянку. НДС не облагается. ",  
  "kbk ": "78011302992029003130 ",  
  "payeeName ": "ГКУ АМПП ",  
  "amount ": 1090000,  
  "payerName ": "Физические лица ",  
  "additionalDataDiscountSize ": "25 ",  
  "additionalDataDiscountDate ": "2025-05-12 ",  
  "msgType ": 4  
}
```

### МАДИ

```
{  
  "subscribe ": "2400000000009908284333643 ",  
  "payerDoc ": {  
    "code ": "200 ",  
    "value ": "7730050504773001001 "  
  },  
  "depType ": "gibdd ",  
  "supplierBillID ": "0356043010825050700027843 ",  
  "billDate ": "2025-05-07T00:00:00+03:00 ",
```

```

    "amountToPay ": 1000000,
    "purpose ": "Оплата штрафа по постановлению 0356043010825050700027843 от
07.05.2025 ",
    "kbc ": "78211602010020001140 ",
    "payeeName ": "Московская административная дорожная инспекция ",
    "amount ": 1000000,
    "payerName ": "- ",
    "additionalDataDiscountSize ": "25 ",
    "additionalDataDiscountDate ": "2025-06-06 ",
    "msgType ": 3
}

```

### 2.3.3.2 Алгоритмы привязки методов оплат

#### Логика работы привязок на Витрине (UI)

##### Оплата Банковской картой с привязкой

Для предоставления Пользователю возможности привязать при оплате Банковскую карту, необходимо в запросе на генерацию ссылки на оплату передать следующие параметры:

Название	Значение	Обязательность	Описание
payerID	string	1	Идентификатор Пользователя в информационной системе Партнера. Длина 28 символов (латинских букв/цифр).

**Важно!** Возможность подписки должна быть согласована с персональным менеджером.

Если данные условия были соблюдены, на UI-форме отобразится чекбокс для привязки Банковской карты.

##### Запрос

```

curl -X POST --location "https://demopay.***.ru/api/multiserv/2.0/pay" \
  -H "Content-Type: application/json; charset=utf-8" \
  -H "Authorization: Bearer testservice" \

```

```
-d '{
  "orderID": "test-045",
  "depType": "merchant122211",
  "defPayType": "card",
  "kvit": true,
  "amount": 400,
  "payerParams": {
    "email": "test@example.tst"
  },
  "paymentParams": {
    "purpose": "Подписка на 1 месяц."
  },
  "payerID": "test09855",
  "returnUrl": "test://abc/",
  "sign": "{{hash}}"
}'
```

### Оплата по СБП с привязкой

Для предоставления Пользователю возможности привязать при оплате свой банковский счёт к СБП, необходимо в запросе на генерацию ссылки на оплату передать следующие параметры:

Название	Значение	Обязательность	Описание
sbpSubscription	object	1	Привязка счета при оплате через СБП, для последующей рекуррентной оплаты без акцепта плательщика (метод sbpRecurrent). payerID должен быть заполнен.
enable	boolean	1	Признак активации привязки счета СБП. Для привязки счета к payerID, значения должно быть передано как "true".
purpose	string	0..1	Назначение подписки, передаётся при оформлении подписки в СБП.

payerID	string	1	Идентификатор Пользователя в информационной системе Партнера. Длина 28 символов (латинских букв/цифр).
---------	--------	---	---

**Важно!** Возможность подписки должна быть согласована с персональным менеджером.

На UI-форме нет чекбокса о согласии на осуществление привязки банковского счета к СБП. Если необходимые условия были соблюдены, соответствующий чекбокс появится в банковском приложении Пользователя.

### Запрос

```
curl -X POST --location "https://demopay.***.ru/api/multiserv/2.0/pay" \
-H "Content-Type: application/json; charset=utf-8" \
-H "Authorization: Bearer testservice" \
-d '{
  "orderID": "test-055",
  "depType": "merchant111111",
  "defPayType": "sbp",
  "kvit": true,
  "amount": 300,
  "payerParams": {
    "email": "test@example.tst"
  },
  "paymentParams": {
    "purpose": "Подписка на 1 месяц."
  },
  "payerID": "test09855",
  "sbpSubscription": {
    "enable": true,
    "purpose": "Автоматическое продление подписки."
  },
  "returnUrl": "test://abc/",
  "sign": "{{hash}}"
}
```

```
}
```

## Бизнес-логика Привязок для API

### Привязка банковского счёта к СБП при выполнении оплаты платежа

Осуществляется методом pay. Для этого, в методе должны быть переданы следующие параметры:

Название	Значение	Обязательность	Описание
sbpSubscription	object	1	Привязка счёта при оплате через СБП, для последующей рекуррентной оплаты без акцепта плательщика (метод sbpRecurrent). payerID должен быть заполнен.
enable	boolean	1	Признак активации привязки счёта СБП. Для привязки счёта к payerID, значения должно быть передано как "true".
purpose	string	0..1	Назначение подписки, передаётся при оформлении подписки в СБП.
payType	string	1	Способ оплаты, должен быть передан как "sbp".
payerID	string	1	Идентификатор Пользователя в информационной системе Партнера. Длина 28 символов (латинских букв/цифр).

### Запрос

```
curl -X POST --location "https://demopay.***.ru/api/multiserv/2.0/pay" \  
-H "Content-Type: application/json; charset=utf-8" \  
-H "Authorization: Bearer testservice" \  
-d '{
```

```

"orderID": "test-055",
"depType": "merchant111111",
"defPayType": "sbp",
"kvit": true,
"amount": 300,
"payerParams": {
  "email": "test@example.tst"
},
"paymentParams": {
  "purpose": "Подписка на 1 месяц."
},
"payerID": "test09855",
"sbpSubscription": {
  "enable": true,
  "purpose": "подписка на 1 месяц"
},
"returnUrl": "test://abc/",
"sign": "{{hash}}"
}'

```

### Привязка Банковской карты после успешной оплаты платежа

Алгоритм привязки:

1. Выполнить оплату платежа по API-схеме с помощью метода pay. В запросе на оплату обязательно необходимо передать параметр **PayerId**.
2. Выполнить вызов метода linkCard, передав в запросе полученный на предыдущем этапе идентификатор платежа (**paymentId**) и параметр **PayerId**.

На данный момент нельзя выполнить привязку Банковской карты методом linkCard по платежу, проведённому по схеме оплаты с UI-формой. Для привязки БК методом linkCard платёж должен быть успешно Авторизован.

### Запрос

```

curl -X POST --location "https://demopay.***.ru/api/multiserv/2.0/linkCard" \
  -H "Content-Type: application/json; charset=utf-8" \
  -H "Authorization: Bearer testservice" \
  -d '{

```

```

    "payerID":"user12345",
    "paymentID":"1234567890",
    "cardName":"дебетовая карта"
  }'

```

## Привязка Банковской карты

Алгоритм привязки:

1. Запрашивается адрес страницы для привязки новой Банковской карты. Для этого выполняется вызов метода saveCard. В ответе будет отдана ссылка для перенаправления пользователя на страницу привязки Банковской Карты.
2. После возврата пользователя со страницы привязки Банковской карты, выполнить вызов метода confirmSave.

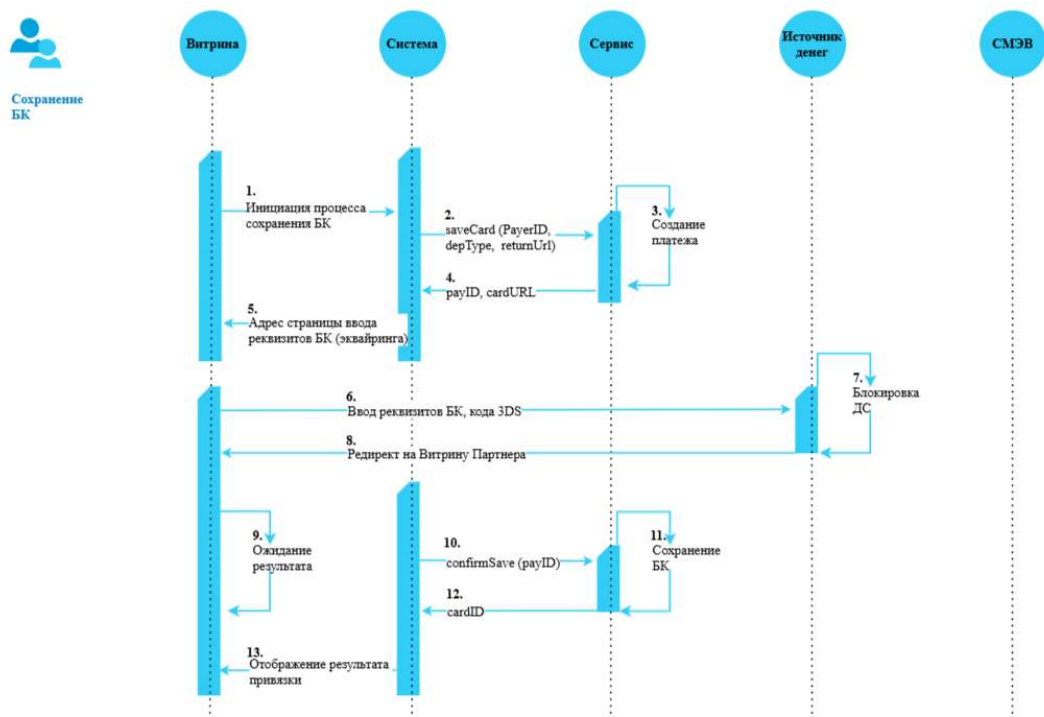


Рисунок 27. Сохранение банковской карты

## Запрос saveCard

```

curl -X POST --location "https://demopay.***.ru/api/multiserv/2.0/saveCard" \
  -H "Content-Type: application/json; charset=utf-8" \
  -H "Authorization: Bearer testservice" \
  -d '{

```

```
"payerID": "user12345",  
"depType": "gibdd",  
"returnUrl": "https://testpartner.ru/e0462fc0872e2/"  
}'
```

### Запрос confirmSave

```
curl -X POST --location "https://demopay.***.ru/api/multiserv/2.0/confirmSave" \  
-H "Content-Type: application/json; charset=utf-8" \  
-H "Authorization: Bearer testservice" \  
-d '{  
    "payID": "511147",  
    "depType": "gibdd"  
}'
```

## 2.4 Модуль «Создания и ведения цифрового профиля пользователя»

Модуль «Создание и ведение цифрового профиля пользователя» реализует следующие функциональные возможности:

- прием запросов от информационной системы партнёра на аутентификацию пользователя с указанием требуемого состава данных;
- перенаправление пользователя на авторизацию в ЕСИА и получение согласия на предоставление данных;
- взаимодействие с ЕСИА для получения персональных данных пользователя (включая паспортные данные, ИНН, СНИЛС, контактную информацию, сведения о транспортных средствах и иные данные);
- обработку, структурирование и передачу полученных данных в систему партнёра по защищённым каналам в формате JSON;
- поддержку различных наборов данных в зависимости от запроса (профиль, документы, транспорт, семейные сведения и др.);
- контроль статусов обработки запросов и обработку ошибок;
- обеспечение гарантированной доставки данных с механизмом повторных попыток передачи;
- поддержку расширяемого формата данных для обеспечения совместимости и развития интеграции.

### 2.4.1 Порядок взаимодействия систем

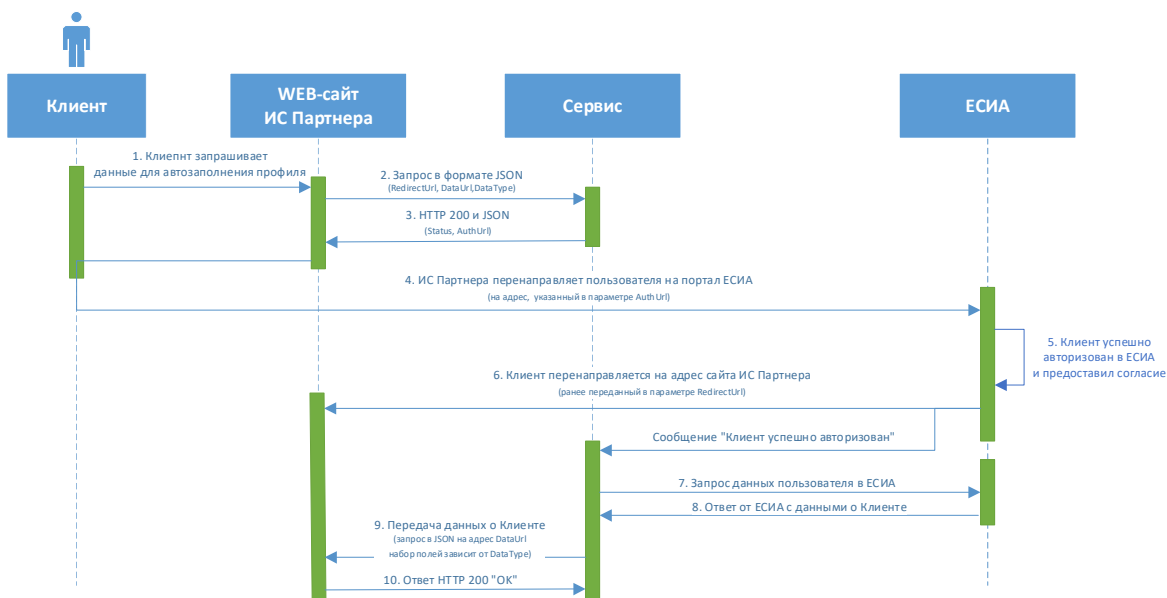


Рисунок 28. Общая схема взаимодействия информационных систем

1. Клиент запрашивает аутентификацию на витрине Партнера.

2. ИС Партнера направляет запрос в Сервис на URL-адрес `https://esia-auth.*/authCP`. Данные в запросе передаются методом POST в формате JSON.

Запрос должен содержать 3 параметра:

- параметр **RedirectUrl** – URL-адрес в ИС Партнера, на который Сервису надо будет перенаправить Клиента после того, как он авторизуется в ЕСИА;
- параметр **DataUrl** – параметр должен содержать URL-адрес в ИС Партнера, на который Сервис должен будет передать данные о Клиенте, полученные из ЦП и уникальный идентификатор запроса, сформированный ИС Партнера.
- Параметр **DataType** – набор запрашиваемых данных по Клиенту, доступны:
  - **passport** – ФИО, паспортные данные;
  - **passportExtDocCont** – ФИО, паспортные данные (расширенные), ИНН, СНИЛС, контактные данные;
  - **vehicleInfo** – информация по транспортным средствам, зарегистрированным на Клиента;
  - **profileCont** – ФИО, адрес, контактные данные;
  - **profileContINN** – ФИО, адрес, контактные данные, дата и место рождения, ИНН;
  - **partnerProfile** – ФИО супруга, адрес, дата заключения брака;
  - **partnerProfileExtINN** – ФИО супруга, адрес, дата заключения брака, дата и место рождения, ИНН.

Пример запроса:

```
curl --location --request POST 'https://esia-auth.*/authCP' --header 'Content-Type: application/json' --data-raw '{
  "RedirectUrl": "https://example.com/auth",
  "DataUrl": "https://example.com/data/ae12fb21e4",
  "DataType": "passportExtDocCont"
}'
```

Сервис обрабатывает полученный запрос и выдает описанные ниже варианты ответов.

3. Сервис выдает ответ HTTP 200 и возвращает в ответ JSON, со следующей структурой полей:

Структура полей JSON:

- **Status** – код статуса обработки запроса (Int):
  - 0 – запрос был успешно обработан;
  - 2 – при обработке запроса произошла ошибка;

- **AuthUrl** – URL, на который ИС Партнера должна будет перенаправить Клиента для выполнения авторизации в ЕСИА (String).
4. WEB-сайт Партнера перенаправляет Клиента на портал ЕСИА (302 redirect), на адрес, указанный в параметре AuthUrl (см. п.3.)
  5. Клиент авторизуется в ЕСИА и дает согласие на предоставление данных.
  6. После того, как Клиент авторизовался в ЕСИА – ЕСИА перенаправляет Клиента на URL, который был передан Сервису от ИС Партнера в параметре RedirectUrl (см. п.2) и уведомляет Сервис о том, что Клиент успешно авторизовался в ЕСИА.
  7. Сервис направляет запрос в ЕСИА для получения данных о Клиенте.
  8. Сервис получает ответ от ЕСИА с данными о Клиенте.
  9. Сервис передает данные о Клиенте в ИС Партнера путем подачи HTTPS-запроса на ранее указанный партнером URL-адрес (параметр DataUrl). Данные передаются в формате JSON.
  10. ИС Партнера получает данные и возвращает Сервису в ответ HTTP 200 и текст «ОК».

#### 2.4.2 Структура полей JSON с данными о Клиенте

- **userData** – набор данных Клиента, полученных из ЕСИА;
- **error** – используется для передачи информации об ошибках (при наличии таковых):
  - **errorID** – код ошибки (String);
  - **errorText** – текст ошибки (String).

ИС Партнера должна корректно обрабатывать ситуации, когда в структуре полей присутствуют прочие поля, явно не перечисленные в расположенных ниже таблицах, или отсутствуют какие-либо из перечисленных. Это необходимо для обеспечения возможности расширения протокола.

Также рекомендовано контролировать параметр `trusted`. Для `trusted=false` может отсутствовать часть обязательных параметров.

## 2.4.2.1 Описание наборов данных Клиента (UserData)

### 2.4.2.1.1 UserData для passport

Таблица 9. Описание полей UserData для passport

№ пп	Код поля	Описание поля	Требования к заполнению	Комментарий
1	series	Серия документа	обязательно	Строка
2	number	Номер документа	обязательно	Строка
3	lastName	Фамилия пользователя	обязательно	Строка
4	firstName	Имя пользователя	обязательно	Строка
5	middleName	Отчество пользователя	необязательно	Строка
6	trusted	Подтвержденная запись	обязательно	Bool

### 2.4.2.1.2 UserData для passportExtDocCont

Таблица 10. Описание полей UserData для passportExtDocCont

№ пп	Код поля	Описание поля	Требования к заполнению	Комментарий
1	oid	Идентификатор пользователя	обязательно	Строка
2	id	Уникальный идентификатор документа	обязательно	Строка
3	status	Статус достоверности	обязательно	Принимает значения: – verified_by_validate; – verified_by_request; – verified_by_push; – unverified.
4	relevance	Релевантность документа	обязательно	Принимает значения: – actual; – expired; – invalid.
5	series	Серия документа	обязательно	Строка
6	number	Номер документа	обязательно	Строка
7	issueDate	Дата выдачи документа	обязательно	Дата в формате: ДД.ММ.ГГГГ
8	type	Тип документа	обязательно	Для паспорта РФ принимает значение RF PASSPORT
9	departmentDoc	Ведомство, к которому относится документ	необязательно	Строка
10	receiptDocDate	Дата и время получения документа в формате TIMESTAMP	необязательно	Задается как количество миллисекунд, прошедших с 00:00:00 UTC 1 января 1970 года
11	validateDateDoc	Дата и время валидации документа в формате TIMESTAMP	необязательно	Задается как количество миллисекунд, прошедших с 00:00:00 UTC 1 января 1970 года
12	lastName	Фамилия пользователя	обязательно	Строка
13	firstName	Имя пользователя	обязательно	Строка
14	middleName	Отчество пользователя	необязательно	Строка
15	gender	Половая принадлежность	обязательно	Принимает значения: – М – мужской пол; – F – женский пол.

16	birthDate	Дата рождения	обязательно	Дата в формате: ДД.ММ.ГГГГ
17	birthplace	Место рождения	необязательно	Строка
18	issuedBy	Кем выдан паспорт	необязательно	Строка
19	issueId	Код подразделения	обязательно	Строка
20	snils	СНИЛС	обязательно	Строка
21	inn	ИНН	обязательно	Строка
22	mobilePhone	Номер мобильного телефона	необязательно	Строка
23	email	Адрес электронной почты	необязательно	Строка
24	homeAddress	Домашний адрес	необязательно	Строка
25	registrationAddress	Адрес места жительства на территории РФ	необязательно	Строка
26	trusted	Подтвержденная запись	обязательно	Bool

### 2.4.2.1.3 UserData для vehicleInfo

Таблица 11. Описание полей UserData для vehicleInfo

№ пп	Код поля	Описание поля	Требования к заполнению	Комментарий
1	oid	Идентификатор пользователя	обязательно	Строка
2	id	Уникальный идентификатор документа	обязательно	Строка
3	status	Статус достоверности	обязательно	Принимает значения*: – verified_by_validate; – verified_by_request; – verified_by_push; – unverified
4	relevance	Релевантность документа	обязательно	Принимает значения: – actual; – expired; – invalid
5	type	Тип документа	обязательно	Принимает значение: VEHICLE_INFO
6	departmentDoc	Ведомство, к которому относится документ	обязательно	Принимает значение «Управление ГИБДД ГУ МВД России»
7	version	Версия документа	обязательно	Принимает значение: «1»
8	createdOn	Дата и время создания документа в формате TIMESTAMP	обязательно	Задается как количество миллисекунд, прошедших с 00:00:00 UTC 1 января 1970 года
9	updatedOn	Дата и время обновления документа в формате TIMESTAMP	обязательно	Задается как количество миллисекунд, прошедших с 00:00:00 UTC 1 января 1970 года
10	receiptDocDate	Дата и время получения документа в формате TIMESTAMP	обязательно	Задается как количество миллисекунд, прошедших с 00:00:00 UTC 1 января 1970 года

11	validateDateDoc	Дата и время валидации документа в формате TIMESTAMP	обязательно, в случае получения выписки	Задается как количество миллисекунд, прошедших с 00:00:00 UTC 1 января 1970 года
12	nameDoc	Наименование документа	обязательно, в случае получения выписки	Строка
13	content	Выписка о транспортном средстве по владельцу в XML, закодированная в формате BASE64	обязательно, в случае получения выписки	Строка, содержащая закодированный в формате BASE64 файл
14	error	Информация об ошибке, предыдущие сведения не затираются и отображаются (при наличии). Если в ответе имеется указанное поле, полученные данные принято считать не актуальными	необязательно	Строка
15	errorUpdatedOn	Дата последнего ответа с ошибкой (метка времени в unixtimestamp). Если в ответе имеется указанное поле, полученные данные принято считать не актуальными	необязательно	Целое число

\* – Поле принимает значения:

- verified\_by\_validate сведения, введённые пользователем и проверенные в ИС поставщика данных;
- verified\_by\_request сведения, полученных из ИС поставщика данных по запросу (в данном случае - система Цифрового профиль является инициатором запроса на получение сведений из ИС поставщика данных по определенному идентификатору);
- verified\_by\_push сведения, полученные из ИС поставщика данных в режиме «push» (в данном случае система ИС поставщика данных является инициатором отправки информации об изменении сведений);
- unverified сведения, проверка по которым в ИС поставщика данных не осуществляется или завершилась ошибкой.

#### 2.4.2.1.4 UserData для ProfileCont

Таблица 12. Описание полей UserData для ProfileCont

№ пп	Код поля	Описание поля	Требования к заполнению	Комментарий
1	oid	Идентификатор пользователя	обязательно	Строка
2	lastName	Фамилия пользователя	обязательно	Строка

3	firstName	Имя пользователя	обязательно	Строка
4	middleName	Отчество пользователя	необязательно	Строка
5	email	Адрес электронной почты	необязательно	Строка
6	mobilePhone	Номер мобильного телефона	необязательно	Строка
7	homeAddress	Домашний Адрес	необязательно	Объект
8	registrationAddress	Адрес места жительства на территории РФ	необязательно	Объект
9	trusted	Подтвержденная запись	обязательно	Bool

#### 2.4.2.1.5 UserData для ProfileContINN

Таблица 13. Описание полей UserData для ProfileContINN

№ пп	Код поля	Описание поля	Требования к заполнению	Комментарий
1	oid	Идентификатор пользователя	обязательно	Строка
2	lastName	Фамилия пользователя	обязательно	Строка
3	firstName	Имя пользователя	обязательно	Строка
4	middleName	Отчество пользователя	необязательно	Строка
5	email	Адрес электронной почты	необязательно	Строка
6	mobilePhone	Номер мобильного телефона	необязательно	Строка
7	homeAddress	Домашний Адрес	необязательно	Строка
8	registrationAddress	Адрес места жительства на территории РФ	необязательно	Строка
9	birthDate	Дата рождения	необязательно	Строка
10	birthPlace	Место Рождения	необязательно	Строка
11	inn	ИНН	необязательно	Строка
12	trusted	Подтвержденная запись	обязательно	Bool

#### 2.4.2.1.6 UserData для PartnerProfile

Таблица 14. Описание полей UserData для PartnerProfile

№ пп	Код поля	Описание поля	Требования к заполнению	Комментарий
1	oid	Идентификатор пользователя	обязательно	Строка
2	id	Уникальный идентификатор документа	обязательно	Строка
3	status	Статус достоверности	– обязательно	Принимает значения *: – verified_by_validate; – verified_by_request; – verified_by_push; – unverified.
4	relevance	Релевантность документа	обязательно	Принимает значения: – actual; – expired; – invalid.
5	issueDate	Дата выдачи документа	обязательно	Дата в формате: ДД.ММ.ГГГГ

6	actDate	Дата составления записи АГС	необязательно	Дата в формате: ДД.ММ.ГГГГ
7	lastNameBefore	Фамилия до брака	необязательно	Строка
8	lastNameAfter	Фамилия после брака	необязательно	Строка
9	firstName	Имя пользователя	необязательно	Строка
10	middleName	Отчество пользователя	необязательно	Строка
11	homeAddress	Сведения о месте жительства	необязательно	Объект
12	addressInRussia	Адрес места жительства на территории РФ	необязательно	Объект

\* – Поле принимает значения:

- verified\_by\_validate сведения, введённые пользователем и проверенные в ИС поставщика данных;
- verified\_by\_request сведения, полученных из ИС поставщика данных по запросу (в данном случае - система Цифрового профиль является инициатором запроса на получение сведений из ИС поставщика данных по определенному идентификатору);
- verified\_by\_push сведения, полученные из ИС поставщика данных в режиме «push» (в данном случае система ИС поставщика данных является инициатором отправки информации об изменении сведений);
- unverified сведения, проверка по которым в ИС поставщика данных не осуществляется или завершилась ошибкой.

#### 2.4.2.1.7 UserData для PartnerProfileExtINN

Таблица 15. Описание полей UserData для PartnerProfileExtINN

№ пп	Код поля	Описание поля	Требования к заполнению	Комментарий
1	oid	Идентификатор пользователя	обязательно	Строка
2	id	Уникальный идентификатор документа	обязательно	Строка
3	status	Статус достоверности	обязательно	Принимает значения *: <ul style="list-style-type: none"> <li>– verified_by_validate;</li> <li>– verified_by_request;</li> <li>– verified_by_push;</li> <li>– unverified.</li> </ul>
4	relevance	Релевантность документа	обязательно	Принимает значения: <ul style="list-style-type: none"> <li>– actual;</li> <li>– expired;</li> <li>– invalid.</li> </ul>
5	issueDate	Дата выдачи документа	обязательно	Дата в формате: ДД.ММ.ГГГГ
6	actDate	Дата составления записи АГС	необязательно	Дата в формате: ДД.ММ.ГГГГ
7	lastNameBefore	Фамилия до брака	необязательно	Строка
8	lastNameAfter	Фамилия после брака	необязательно	Строка
9	firstName	Имя пользователя	необязательно	Строка
10	middleName	Отчество пользователя	необязательно	Строка
11	homeAddress	Сведения о месте жительства	необязательно	Объект

12	addressInRussia	Адрес места жительства на территории РФ	необязательно	Объект
13	birthDate	Дата рождения	обязательно	Дата в формате: ДД.ММ.ГГГГ
14	birthplace	Место рождения	необязательно	Объект
15	inn	ИНН	обязательно	Строка

\* – Поле принимает значения:

- verified\_by\_validate сведения, введённые пользователем и проверенные в ИС поставщика данных;
- verified\_by\_request сведения, полученных из ИС поставщика данных по запросу (в данном случае - система Цифрового профиль является инициатором запроса на получение сведений из ИС поставщика данных по определенному идентификатору);
- verified\_by\_push сведения, полученные из ИС поставщика данных в режиме «push» (в данном случае система ИС поставщика данных является инициатором отправки информации об изменении сведений);
- unverified сведения, проверка по которым в ИС поставщика данных не осуществляется или завершилась ошибкой.

#### 2.4.2.2 Повторная отправка данных

Если по каким-либо причинам ИС Партнера не подтвердила получение данных – то Сервис будет пытаться передать данные в ИС Партнера еще несколько раз. Данные передаются:

- непосредственно в момент, когда Сервис получил данные о Клиента;
- через 30 секунд;
- через 60 секунд;
- через 5 минут;
- через 10 минут;
- через 20 минут;
- через 30 минут.

После последней попытки передачи данные о Клиенте считаются устаревшими и Сервис не производит дальнейших попыток их передачи в ИС Партнера.

#### 2.4.2.3 Используемые коды ошибок

В случае возникновения ошибок Сервис в формате JSON возвращает в ИС Партнера код ошибки и текст ошибки (поля errorID и errorText).

Таблица 16. Используемые коды ошибок и их описания

Код ошибки	Описание ошибки
ESIA-000001	Внутренняя ошибка
ESIA-005003	Доступ запрещен
ESIA-005013	SecurityErrorEnum.expiredToken
ESIA-005029	SecurityErrorEnum.scopeNotAllowed
ESIA-020027	ПЭП не найдена
ESIA-020203	Документ не найден
ESIA-020131	Учетная запись не найдена, либо не предоставлен верифицированный номер ИНН.
ESIA-030003	Неверно указаны параметры
ESIA-035000	Пользователь не найден
ESIA-036701	Не указана мнемоника типа согласия
ESIA-036701	Не найден тип согласия
ESIA-036703	Указанные скоупы выходят за рамки разрешенных для типа согласия
ESIA-036704	Запрещено указывать скоупы для типа согласия
ESIA-036705	Необходимо указать хотя бы одно действие
ESIA-036706	Указанное действие не существует
ESIA-036707	Необходимо указать хотя бы одну цель
ESIA-036710	Согласие не найдено
ESIA-036716	Указано некорректное время истечения срока действия согласия
ESIA-036718	Необходимо предварительно выдать согласие Министерству цифрового развития, связи и массовых коммуникаций Российской Федерации
ESIA-036726	Указанная цель не существует
ESIA-036727	Необходимо указать одну цель согласия
ESIA-036729	Архив с подписанным согласием не может быть получен, т.к. произошла ошибка на этапе выгрузки архива в хранилище
ESIA-038006	В запросе отсутствует ЭЦП организации
ESIA-039230	Недостаточно прав для обновления лицензии
ESIA-039870	Выбранная информационная система не найдена
ESIA-058012	Данный пользователь не является руководителем ИП/ЮЛ

## 2.4.3 Примеры информационного обмена

### 2.4.3.1 Пример userData JSON для passport

```
{
  "lastName": "Петров", "firstName": "Иван", "middleName": "Сергеевич",
  "series": "0001", "number": "393393", "trusted": true
}
```

### 2.4.3.2 Пример userData JSON для passportExtDocCont

```
{
  "oid": "1000376121", "id": "67664", "receiptDocDate": 1576835468000,
  "validateDateDoc": 1576835468000, "relevance": "actual", "status":
  "verified_by_validate", "lastName": "Петров", "firstName": "Иван",
  "middleName": "Сергеевич", "birthPlace": "Курская область", "gender":
  "М", "birthDate": "23.06.1989", "departmentDoc": "Главное управление №1
  по вопросам миграции МВД России", "series": "0001", "number": "393393",
  "issueId": "145854", "issuedBy": "РОВД 89", "issueDate": "17.07.2009",
  "type": "RF_PASSPORT", "snils": "204-585-302 46", "inn": "403956183997",
  "mobilePhone": "+7(905)6578090", "homeAddress":{ "addressStr":
  "Кемеровская Область, Таштагольский Район, Шерегеш Поселок городского
  типа", "region": "23", "flat": "пом.419", "countryId": "RUS", "house":
  "43Ke", "frame": "2Lk", "zipCode": "394000", "building": "2f", "street":
```

```
"Советская Улица", "fiasCode": "720b25da-f43e-4204-9013-3cb06be3e9e4"},
"trusted": true
}
```

### 2.4.3.3 Пример userData JSON для vehicleInfo

```
{
  "oid": "1077130925", "id": "1ab0a427-48e9-47f7-ad10-6b58ff3d75e0",
  "version": 1, "createdOn": 1616081460770, "updatedOn": 1616081460770,
  "receiptDocDate": 1616081460770, "validateDateDoc": 1616081460601,
  "relevance": "actual", "status": "verified_by_request", "departmentDoc":
  "Управление ГИБДД ГУ МВД России", "nameDoc": "Выписка о ТС по
  владельцу", "content": "PD94b....JzPg==", "type": "VEHICLE_INFO"
}
```

Пример раскодированного содержимого поля content:

```
{
  "ns1:CarcaseColor": "СИНИЙ",
  "ns1:CarcaseNumber": "XWEFX411AE000**04",
  "ns1:Category": "B",
  "ns1:DriveUnitType": "2",
  "ns1:EngineModel": "G4FG",
  "ns1:EngineNum": "EH71**54",
  "ns1:EnginePowerHorse": "130.0",
  "ns1:EnginePowerkVt": "95.3",
  "ns1:EngineVolume": "1591",
  "ns1:GovRegNumber": "C7**OY178",
  "ns1:ManufacturedYear": "2014",
  "ns1:MaxWeight": "1720",
  "ns1:ModelMarkName": "КИА CERATO YD",
  "ns1:OwnerPeriods": {
    "ns1:Owner": {
      "ns1:DateStart": "2014-12-01",
      "ns1:OwnerType": "2"
    }
  },
  "ns1:Owners": {
    "ns1:Individual.IP": {
      "ns1:BirthDay": "198*-06-**",
      "ns1:BirthPlace": "ЛЕНИНГРАД",
      "ns1:DocumentDate": "2008-07-21",
      "ns1:DocumentNumSer": "номер паспорта",
      "ns1:FirstName": "ПАВЕЛ",
      "ns1:IDDocumentType": "5",
      "ns1:LastName": "С***",
      "ns1:MiddleName": "СЕРГЕЕВИЧ",
      "ns1:RegAddress": "адрес"
    },
    "ns1:OwnerLegalType": "Ind"
  },
  "ns1:PTS": {
    "ns1:PTSIssueAgency": "ООО",
    "ns1:PTSNum": "39HY07**24",
    "ns1:PTSRegDate": "2014-06-10",
    "ns1:PTSSpecialTags": "01.12.2014 ДКП N001260 ОТ 21.11.2014 VIN
    ЗАВОДА-ИЗГОТ.KNAFX411AE5876963*УТИЛ СБОР ИНН3906072056",
    "ns1:PTSType": "ПТС"
  },
}
```

```

    "ns1:RecordStatus": "1",
    "ns1:RegActions": {
      "ns1:RegAction": {
        "ns1:LeasingFlag": "false",
        "ns1:RegActionName": "Новых, произведенных в России или
везенных, а также ввезенных в Россию бывших в эксплуатации",
        "ns1:RegDate": "2014-12-01"
      }
    },
    "ns1:RegistrationDoc": {
      "ns1:RegDocDocumentDate": "2014-12-01",
      "ns1:RegDocSeriesNumber": "522522**82",
      "ns1:RegDocSpecialTags": None,
      "ns1:RegDocumentType": "СТС"
    },
    "ns1:RestrictionsFlag": "false",
    "ns1:SearchingSpec": {
      "ns1:SearchingSpecFlag": "false"
    },
    "ns1:SearchingTransportFlag": "false",
    "ns1:UniqueRowNumber": "658893626",
    "ns1:VIN": "XWEFX411AE000**04",
    "ns1:VenicleType": "СЕДАН",
    "ns1:WeightWithoutLoading": "1338",
    "ns1:WheelLocation": "1"
  }
}

```

#### 2.4.3.4 Пример userData JSON для ProfileCont

```

{
  "oid": "1000376121", "lastName": "Петров", "firstName": "Иван",
  "middleName": "Сергеевич", "email": "123email@123.ru", "mobilePhone":
  "+7(905)6578090", "homeAddress":{ "addressStr": "Кемеровская Область,
Таштагольский Район, Шерегеш Поселок городского типа", "region": "23",
"flat": "пом.419", "countryId": "RUS", "house": "43Ke", "frame": "2Lk",
"zipCode": "394000", "building": "2f", "street": "Советская Улица",
"fiasCode": "720b25da-f43e-4204-9013-3cb06be3e9e4"}
  "trusted": true
}

```

#### 2.4.3.5 Пример userData JSON для ProfileContINN

```

{
  "oid": "1000376121", "lastName": "Петров", "firstName": "Иван",
  "middleName": "Сергеевич", "birthPlace": "Курская область", "birthDate":
  "23.06.1989", "inn": "403956183997", "email": "123email@123.ru",
  "mobilePhone": "+7(905)6578090", "homeAddress":{ "addressStr":
  "Кемеровская Область, Таштагольский Район, Шерегеш Поселок городского
  типа", "region": "23", "flat": "пом.419", "countryId": "RUS", "house":
  "43Ke", "frame": "2Lk", "zipCode": "394000", "building": "2f", "street":
  "Советская Улица", "fiasCode": "720b25da-f43e-4204-9013-3cb06be3e9e4"}
  "trusted": true
}

```

#### 2.4.3.6 Пример userData JSON для PartnerProfile

```
{ "oid": "1077134819", "id": "0b6d68cb-80f4-4d76-9941-5b9b513656f2",  
  "relevance": "actual", "status": "verified_by_request", "issueDate":  
  "25.03.2021", "actDate": "25.03.2021", "firstName": "Марина", "middleName":  
  "Петровна", "lastNameBefore": "Колодова", "lastNameAfter": "Власова",  
  "homeAddress": { "addressInRussia": { "addressFiasStr": "Российская  
  Федерация, г. Москва, ул. Планерная, д.56", "oktmo": "45369000",  
  "fiasCode": "cf31b1d0-fad0-4723-84a9-a582e653e051", "regionCode": "77",  
  "regionName": "Нет данных", "house": "56"}  
}
```

#### 2.4.3.7 Пример userData JSON для PartnerProfileExtINN

```
{ "oid": "1077134819", "id": "0b6d68cb-80f4-4d76-9941-5b9b513656f2",  
  "relevance": "actual", "status": "verified_by_request", "issueDate":  
  "25.03.2021", "actDate": "25.03.2021", "firstName": "Марина", "middleName":  
  "Петровна", "lastNameBefore": "Колодова", "lastNameAfter": "Власова",  
  "birthDate": "01.01.1992", "birthPlace": { "placeSign": "3", "placeStr":  
  "Российская Федерация, г. Москва", "countryName": "Российская федерация",  
  "oktmo": "45000000" }, "homeAddress": { "addressInRussia": {  
  "addressFiasStr": "Российская Федерация, г. Москва, ул. Планерная, д.56",  
  "oktmo": "45369000", "fiasCode": "cf31b1d0-fad0-4723-84a9-a582e653e051",  
  "regionCode": "77", "regionName": "Нет данных", "house": "56" } }, "inn":  
  "403956183997"  
}
```