|  |
| --- |
| **Payguide Acquirer  Описание взаимодействия с магазином** |

КОНТРОЛЬ ВЕРСИЙ ДОКУМЕНТА

|  |  |  |  |
| --- | --- | --- | --- |
| Ссылка на документ | |  | |
| Описание документа | | Документ содержит сценарии выполнения платежей и других операций, а также описание протоколов взаимодействия между магазином и PGA. | |
| Версия документа | Дата | Создан / изменен | Внесенные изменения |
| 1.36 | 17.10.2023 | Ерчиковская Е.В. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.35 | 06.10.2023 | Ерчиковская Е.В.  Емельянов А.В. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.34 | 04.10.2023 | Скрежендевский В.В. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.33 | 29.09.2023 | Ерчиковская Е.В. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.32 | 06.09.2023 | Ягур В.С.  Яковцев А.Л. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.31 | 05.09.2023 | Ягур В.С. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.30 | 14.08.2023 | Ягур В.С. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.29 | 04.08.2023 | Муравицкая А.В.  Ягур В.С. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.28 | 23.06.2023 | Муравицкая А.В.  Ягур В.С. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.27 | 14.06.2023 | Муравицкая А.В. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.26 | 13.06.2023 | Муравицкая А.В. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.25 | 05.06.2023 | Муравицкая А.В. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.24 | 01.06.2023 | Муравицкая А.В. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.23 | 25.05.2023 | Муравицкая А.В. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.22 | 18.04.2023 | Муравицкая А.В. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.21 | 20.12.2022 | Скрежендевский В.В. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.20 | 27.10.2022 | Скрежендевский В.В. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.19 | 05.10.2022 | Скрежендевский В.В. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.18 | 03.10.2022 | Ерчиковская Е.В. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.17 | 28.09.2022 | Скрежендевский В.В. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.16.2 | 13.09.2022 | Скрежендевский В.В. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.16 | 08.09.2022 | Ерчиковская Е.В. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.15 | 15.12.2021 | Анискин Р.А. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.14 | 09.12.2021 | Анискин Р.А. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.13 | 22.09.2021 | Мамченкова М.А. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.12 | 07.06.2021 | Мамченкова М.А. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.11 | 28.04.2021 | Анискин Р.А. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.10 | 02.09.2020 | Анискин Р.А. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.9 | 08.06.2020 | Мамченкова М.А. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.8 | 27.01.2020 | Анискин Р.А. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.7 | 12.12.2019 | Мамченкова М.А. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.6 | 06.12.2019 | Мамченкова М.А. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.5 | 30.10.2019 | Анискин Р.А. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.4 | 19.08.2019 | Анискин Р.А. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.3 | 03.07.2019 | Анискин Р.А. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.2 | 12.04.2019 | Тарасенко Я.Н. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.1 | 14.12.2018 | Летунович Ю.Е. | См. [Внесенные изменения](#_Внесенные_изменения) |
| 1.0 | 02.10.2017 | Зенкевич А.А. | Начальная версия документа. |

ПЕРЕЧЕНЬ ДОКУМЕНТОВ, НА КОТОРЫЙ ССЫЛАЕТСЯ ДАННЫЙ ДОКУМЕНТ

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Название документа | Версия, Дата | Автор |
| [1] | Payguide Acquirer 4 Руководство пользователя консоли магазина | 1.9,  25.01.2023 | Мамченкова М.А. |
| [2] | Payguide Acquirer 4 Протокол OpenAPI Проведение платежей | 1.38,  04.08.2023 | Анискин Р.А. |

ДОКУМЕНТ РАСПРОСТРАНЯЕТСЯ В СООТВЕТСТВИИ С НИЖЕПРИВЕДЕННОЙ ТАБЛИЦЕЙ:

|  |  |
| --- | --- |
| Имя | Компания, должность |
|  |  |
|  |  |

Содержание

[Введение 6](#_Toc148447019)

[1.1 Цель документа 6](#_Toc148447020)

[1.2 Внесенные изменения 6](#_Toc148447021)

[1.1 Термины и сокращения 7](#_Toc148447022)

[2 Архитектура 9](#_Toc148447023)

[2.1 Архитектура решения 9](#_Toc148447024)

[3 Подключение магазина к PGA 10](#_Toc148447025)

[4 Сценарии платежа 11](#_Toc148447026)

[4.1 Сценарий выполнения платежа 11](#_Toc148447027)

[4.2 Сценарий выполнение платежа, инициированного клиентом, без повторного ввода параметров карты 12](#_Toc148447028)

[4.3 Сценарий выполнения рекуррентного платежа, инициированного магазином 13](#_Toc148447029)

[4.4 Сценарий проведения платежа Газпромпей с определения средства платежа на платежной странице 15](#_Toc148447030)

[4.5 Сценарий проведения платежа Газпромпей с определения средства платежа на стороне ТСП 17](#_Toc148447031)

[4.6 Сценарий проведения In-App платежа Mir Pay для МП ТСП с отменой 19](#_Toc148447032)

[4.7 Сценарий проведения In-App платежа Mir Pay для ТСП с библиотекой MirPaySDK.aar с отменой 22](#_Toc148447033)

[4.8 Сценарий проведения AFT операций с вводом данных карты на платежной странице 24](#_Toc148447034)

[4.9 Сценарий проведения AFT операций по ранее сохраненной карте 25](#_Toc148447035)

[4.10 Сценарий проведения OCT операций с вводом данных карты на платежной странице 27](#_Toc148447036)

[4.11 Сценарий проведения OCT операций по сохраненной карте 28](#_Toc148447037)

[4.12 Сценарий проведения P2P операций с вводом данных карты на платежной странице. 30](#_Toc148447038)

[4.13 Сценарий проведения P2P операций по сохраненной карте. 31](#_Toc148447039)

[5 Сценарии регистрации карт 32](#_Toc148447040)

[5.1 Сценарий регистрации карты 32](#_Toc148447041)

[6 Протоколы взаимодействия 33](#_Toc148447042)

[6.1 Константы 33](#_Toc148447043)

[6.2 Инициирование платежа 34](#_Toc148447044)

[6.3 Инициирование платежа по сохранённым реквизитам карты 35](#_Toc148447045)

[6.4 Двухфазное взаимодействие с магазином 37](#_Toc148447046)

[6.4.1 Проверка доступности платежа в магазине 38](#_Toc148447047)

[6.4.1.1 Передача параметров подмагазина 40](#_Toc148447048)

[6.4.2 Регистрация платежа в магазине 41](#_Toc148447049)

[6.4.3 Регистрация карты 43](#_Toc148447050)

[7 Приложение 1 Схема сообщений магазина CPARes, RPRes 45](#_Toc148447051)

[8 Приложение 2 Расширенные коды завершения платежа 47](#_Toc148447052)

Список таблиц, использованных в документе

[Таблица 1 Список изменений, внесенных в документ 6](#_Toc148447053)

[Таблица 2 Термины и сокращения 7](#_Toc148447054)

[Таблица 3 Данные требуемые от магазина 10](#_Toc148447055)

[Таблица 4 Параметры запроса StartPaymentReq 34](#_Toc148447056)

[Таблица 5 Признаки обязательности полей в сообщениях 37](#_Toc148447057)

[Таблица 6 Параметры запроса CPAReq 38](#_Toc148447058)

[Таблица 7 Параметры ответа CPARes 38](#_Toc148447059)

[Таблица 8 Параметры запроса RPReq 41](#_Toc148447060)

[Таблица 9 Параметры ответа RPRes 43](#_Toc148447061)

# Введение

## Цель документа

Данный документ содержит описание сценариев платежей и верификаций карт, а также описание протоколов взаимодействия между магазином и Payguide™ Acquirer, разработанной компанией Intervale.

Документ описывает PGA версии 4.2.86.

## Внесенные изменения

Таблица 1 Список изменений, внесенных в документ

| Версия документа | Раздел документа | Изменение |
| --- | --- | --- |
| 1.36 | 4.7 | Добавлен сценарий проведения In-App платежа для МП ТСП с библиотекой MirPaySDK.aar с отменой. |
| 4.6 | Сценарии проведения MirPay транзакции с использованием платежных страниц PGA и проведения MirPay транзакции со страницы сайта магазина заменены на сценарий проведения In-App платежа для МП ТСП с отменой. |
| 1.35 | 4.12  4.13 | Добавлены сценарии проведения Р2Р операций. |
| 6.4.1 | Добавлено значение P2P для операции P2P. Обновлена Таблица 7. |
| 6.2 | Обновлена Таблица 4. |
| 7 | Добавлено значение P2P для операции P2P. |
|  | Удалена информация об «Apple Pay», «Google Pay». |
| 1.34 | 6.4.1.1 | Добавлен пункт 6.4.1.1. |
| 6.4.1 | Изменена длина поля «submerchant-data.name» (Таблица 7). |
| 1.33 | 6.2 | Добавлен рисунок 19 «Платежная страница». |
| 6.4.3 | Добавлен пункт Регистрация карты. |
| 6.3 | Обновлена информация. |
| 1.32 | 1.1 | Добавлены термины «UCOF», «MIT». |
| 6.4.1 | В Таблица 7 добавлен параметр «card\_on\_file». |
| 1.31 | 6.3 | Обновлен раздел. Добавлены два подтипа для платежей по сохранённым реквизитам карты: MIT, UCOF. |
| 1.30 | 6.2 | В Таблица 4 для параметра src\_type добавлено новое возможное значение: mir\_pay. |
| 6.3 | Добавлена поддержка новых полей recurringExpiry, recurringFrequency для платежей по сохранённым реквизитам карты. |
| 1.29 | 1.1 | В таблицу «Термины и сокращения» добавлен термин «MCC». |
| 1.28 | 6.4.1 | В Таблица 7 «Параметры ответа CPARes» добавлены параметры тега «submerchant-data». Добавлен пример. |
| 1.27 | Приложение 2 | Добавлена ссылка на Приложение 2 Расширенные коды завершения платежа. |
| 0 | Обновлелен пример запроса инициирования платежа (StartPaymentReq). |
| 1.26 |  | Обновлен подраздел: в запросе Pre-Authorization поле 48.66 не передавать, поле 48.42 = 403. |
| 1.25 |  | Обновлен подраздел: поле 48.66 в ответе на запрос Pre-Authorization - необязательное. |
| 1.24 |  | Добавлен подраздел. |
| 1.23 | 6.4.1.1 | В Таблица 8 в параметр p.srcType добавлен тип источника платежа - MIR\_PAY. Добавлен пример регистрации платежа Mir Pay в магазине. |
| 1.22 |  | Поддержка версионности |
| 1.21 | 4.4 | Добавлен раздел «Сценарий проведения платежа Газпромпей с определения средства платежа на платежной странице» |
| 4.5 | Добавлен раздел «Сценарий проведения платежа Газпромпей с определения средства платежа на стороне ТСП» |
|  | Добавлен раздел «Сценарий проведения MirPay транзакции с использованием платежных страниц PGA» |
| 1.20 | 3 | Добавлены требования к сертификатам |
| 1.19 | 0 | Добавлены подразделы 4.9 и 4.11 |
| 6.4.1 | В Таблица 7 добавлен тэг <mir-extension> |
| 1.18 | 6.4.1.1 | Изменено описание параметра p.transmissionDateTime, p.rrn, p.authcode сообщения RPReq |
| 1.17 | 4.10 | Добавлен сценарий OCT перевод по сохраненной карте |
| 4.4 | Добавлен сценарий AFT перевод по сохраненной карте |
| 1.16.2 | 4.10 | Внесены изменения |
| 4.4 | Внесены изменения |
| 1.16 | 4.10 | Добавлен сценарий проведения OCT операций |
| 4.4 | Добавлен сценарий проведения AFT операций |
| 1.15 | 6.3, 6.4 | Добавлены примеры сообщений при рекуррентном платеже |
| 6.4.1.1 | Добавлено описание новых полей в Таблица 8 |
| Приложение 1 | Обновлена схема сообщений CPARes и RPRes |
| 1.14 | 0 | Добавлено описание параметра src.type, дополнено описание параметра paymentId |
| 6.4.1.1 | Добавлено описание новых полей в Таблица 8 |
| 1.13 |  | Актуализирован весь документ |
| 1.12 | 6 | Добавлены примеры для рекуррентных платежей |
| 1.11 | 6.4.1 | Обновлено описание поля purchase.shortDesc в Таблица 7 Параметры ответа CPARes |
| 1.10 | 6.4.1 | Обновлено описание поля transaction-type в Таблица 7 Параметры ответа CPARes |
| 1.9 | 0 | Добавлен параметр paymentId в Таблица 4 Параметры запроса StartPaymentReq |
| 1.8 | 6.4.1.1 | Изменено описание параметра p.transmissionDateTime сообщения RPReq. |
| 1.7 |  | В запросе MerchantApi добавлен запрос по получению информации для сверки |
| 1.6 |  | Поддержка версионности |
|  | Поддержка версионности |
| 1.5 | 6.4 | Изменено описание параметра purchase.account-amount.fee сообщения CPARes, и параметров p.rrn и p.authcode сообщения RPReq. Подробнее описана структура сообщений CPARes и RPRes. Поправлен порядок параметров сообщения RPReq |
| 1.4 | 0 | В запросе инициирования платежа добавлено поле preauthorize |
| 1.3 | 6.4 | Добавлено поле CPARes.submerchant, поправлено поле RPReq.ext\_result\_code |
| 1.2 | [7.1](#_Сценарий_регистрации_карты) | Добавлено описание сценария регистрации карты |
| 1.1 |  | Добавлено описание сценария верификации карт |
| 1.0 |  | Начальная версия документа |

## Термины и сокращения

В данном подразделе определяются термины и сокращения, используемые в документе.

Таблица 2 Термины и сокращения

| Термин | Описание |
| --- | --- |
| **Клиент** | Держатель международной банковской карты, осуществляющий покупки в электронном (интернет) магазине. |
| **ПЦ** | Процессинговый центр. |
| **ТСП/Merchant/Магазин** | Торгово-сервисное предприятие – поставщик товаров или услуг (интернет-магазин). |
| **PGA** | Payguide™ Acquirer – программный комплекс, обеспечивающий проведение безопасных электронных платежей с применением международных банковских карт в адрес интернет-магазинов. |
| **Эквайер** | Банк-эквайер, финансовый институт, обслуживающий торгово-сервисное предприятие. |
| **ЭЦП** | Электронная цифровая подпись – последовательность символов, являющаяся реквизитом электронного документа и предназначенная для подтверждения целостности и подлинности электронного документа. |
| **ACS (Access Control Server)** | Компонент, обеспечивающий проведение аутентификации клиента при осуществлении электронных платежей, сертифицирован в Visa и MasterCard на соответствие протоколу 3-D Secure. |
| **AReq (Authentication Request)** | Сообщение, направляемое от компонента 3DS Server через компонент DS (Directory Server) к компоненту ACS в домене Эмитента, чтобы инициировать в нем процесс Аутентификации. |
| **ARes (Authentication Response)** | Сообщение, возвращаемое от компонента ACS, расположенного в домене Эмитента, через компонент DS компоненту 3DS Server в ответ на сообщение Authentication Request (AReq). |
| **CPAReq (Check Payment Available Request)** | Сообщение, которое содержит запрос на проверку возможности проведения операции магазином в процессе проведения платежной операции. |
| **CPARes (Check Payment Available Response)** | Сообщение, которое содержит результат проверки возможности проведения операции магазином в процессе проведения платежной операции. |
| **CReq (Challenge Request)** | Сообщение, которое направляется от компонента 3DS SDK или от компонента 3DS Server, в котором направляется дополнительная информация, полученная от Держателя карты, компоненту ACS для поддержки процесса аутентификации |
| **CReq (Challenge Response)** | Сообщение, которое направляется от компонента ACS в ответ на сообщение CReq. Оно может содержать результат аутентификации Держателя карты, или, если применяется приложение ТСП со встроенным 3DS SDK, в том числе может содержать признак того, что требуется дополнительное взаимодействие с Держателем карты с целью его аутентификации. |
| **DS (Directory Server)** | Внешний сервис, выполняющий маршрутизацию запросов по аутентификации платежа. |
| **HTTP** | [Протокол](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB) [прикладного уровня](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D1%8B_%D0%BF%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D1%8F) передачи данных. |
| **HTTPS** | Расширение [протокола](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB_%D0%BF%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D1%87%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) [HTTP](https://ru.wikipedia.org/wiki/HTTP), поддерживающее [шифрование](https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5). |
| **MPI (Merchant Plug-In)** | Программное обеспечение, являющееся частью ПО, сертифицированное в VISA на соответствие стандарту «3-D Secure», выполняющее проверку цифровой подписи банка-эмитента при проведении интернет-платежей. |
| **PAReq (Payer Authentication Request)** | Сообщение, которое содержит запрос на аутентификацию плательщика, формируемый магазином и включающий всю необходимую информацию о платеже. |
| **PARes (Payer Authentication Response)** | Сообщение, которое содержит результат аутентификации, получаемый магазином от банка-эмитента. |
| **RPReq (Register Payment Request)** | Запрос регистрации платежа в магазине. |
| **RPRes (Register Payment Response)** | Ответ о регистрации платежа в магазине. |
| **RReq (Result Request)** | Сообщение, направляемое компонентом ACS через компонент DS для передачи результатов операции. |
| **RRes (Result Response)** | Сообщение, направляемое компонентом 3DS Server в адрес компонента ACS через компонент DS для уведомления о получении сообщения Results Request. |
| **SSL (Secure Sockets Layer)** | Криптографический протокол, обеспечивающий безопасную передачу данных по сети. |
| **3-D Secure/3DS** | Протокол аутентификации владельца карты, который используется как дополнительный уровень безопасности для интернет-платежей. Разработан компанией Visa для программы Verified by Visa. Принят компанией MasterCard для программы MasterCard SecureCode. |
| **VEReq (Verify Enrollment Request)** | Сообщение, которое содержит запрос, отправляемый магазином в Directory Server при совершении платежа с целью выяснения участия карты клиента в программе 3-D Secure. |
| **VERes (Verify Enrollment Response)** | Сообщение, которое содержит результат участия карты клиента в программе 3-D Secure, включающее адрес банка-эмитента, способного провести аутентификацию плательщика. |
| **MCC** | Merchant Category Code (MCC) - представляет собой четырехзначное число, описывающие основную деятельность продавца. MCC используются для определения типа бизнеса, которым занимается продавец. |
| **UCOF** | Платеж, запущенный магазином по инициативе держателя карты (по событию). |
| **MIT** | Платеж по инициативе магазина (по расписанию). |



# Архитектура

## Архитектура решения

Система поддержки электронной коммерции представляет собой комплексное решение, которое включает в себя весь набор компонентов, необходимых для проведения безопасных электронных платежей с применением международных банковских карт в адрес интернет-магазинов.



Рисунок 1 Архитектура решения

Архитектурно PGA состоит из следующих компонентов (Рисунок 1):

* Покупатель – клиент интернет-магазина, владеющий банковской картой;
* Магазин – интернет-магазин, товары либо сервисы которого оплачиваются Клиентом;
* PGA – платформа электронной коммерции, предназначенная для проведения удаленных финансовых транзакций;
* ПЦ Эквайера – система, предназначенная для проведения финансовых и сервисных транзакций по картам платежных систем: осуществляет списание денежных средств с карты клиента и перевод на счет магазина;
* Directory Server – компонент платежной системы, предназначенный для хранения информации о всех диапазонах карт участвующих в программе 3-D Secure. Используется для маршрутизации запросов MPI на ACS банка-эмитента;
* ACS Эмитента – компонент, обеспечивающий проведение аутентификации клиента при осуществлении электронных платежей, сертифицирован в Visa и MasterCard на соответствие протоколу 3-D Secure.

# Подключение магазина к PGA

Для подключения к сервисной платформе магазин должен предоставить данные, указанные в Таблица 3.

Таблица 3 Данные требуемые от магазина

| Требуемые данные | Описание | Обязательность |
| --- | --- | --- |
| CheckPaymentAvail URL | Адрес, на который PGA отправляет запрос для проверки возможности проведения платежа в магазине (первая фаза взаимодействия). | Обязательно для основных сценариев платежа. |
| RegisterPayment URL | Адрес, на который PGA отправляет запрос для регистрации результата платежа в магазине (вторая фаза взаимодействия). | Обязательно для основных сценариев платежа. |
| SSL сертификат | SSL сертификат выпущенный центром сертификации. | Обязательно. |
| Наименование магазина | Наименование магазина, отображаемое клиенту на платежной странице. | Обязательно. |
| Данные контактного лица | ФИО, номер телефона и email технического специалиста осуществляющего подключение магазина к Сервисной платформе. | Обязательно. |
| HTTP Authentication Login/HTTP Authentication Password | Технологическая учетная запись (имя пользователя и пароль) для HTTP. | Опционально. |
| fav.ico | Иконка веб-сайта. | Опционально. |
| Логотип 35x35, логотип 59x59, логотип для платежной страницы | Логотипы магазина, отображаемые клиенту на платежной странице в процессе проведения платежа. | Опционально. |

Поле Common Name SSL сертификата магазина должно соответствовать доменному имени сервера в адресах двухфазного взаимодействия (CPAReq/RPReq).

# Сценарии платежа

## Сценарий выполнения платежа

Сценарий проведения обычного платежа представлен на Рисунок 2.

Этот сценарий предусматривает двухфазное взаимодействие с магазином и ввод реквизитов карты клиентом на платёжной странице.



Рисунок 2 Сценарий выполнения платежа

Прямой поток событий сценария выполнения платежа выглядит следующим образом:

* 1. Клиент формирует заказ на сайте интернет-магазина и инициирует оплату сформированного заказа.
  2. Клиент перенаправляется на адрес платежной страницы, содержащий параметры операции и идентификатор магазина.
  3. Платежная страница формирует StartPaymentReq.
  4. PGA передает в магазин запрос проверки возможности проведения платежа (CPAReq).
  5. Магазин возвращает в PGA ответ о возможности проведения операции (CPARes).
  6. PGA отправляет в платежную страницу сообщение с информацией о платеже для подтверждения пользователем.
  7. Платежная страница отображает запрос ввода параметров банковской карты с возможностью привязки карты к магазину.
  8. Клиент вводит реквизиты карты. В PGA направляются параметры карты.
  9. PGA взаимодействует с Directory Server для проверки участия банковской карты в программе 3-D Secure. Успешно.
  10. PGA запрашивает аутентификацию клиента.
  11. Клиент перенаправляется на аутентификацию в ACS Эмитента.
  12. Клиент проходит 3DS-аутентификацию (PAReq).
  13. PGA через браузер получает от сервера аутентификации банка-эмитента сообщение с результатом аутентификации (PARes).
  14. PGA направляет в ПЦ банка-эквайера запрос авторизации на сумму покупки. ПЦ банка-эквайера возвращает результат авторизации в PGA.
  15. PGA направляет в магазин запрос на регистрацию платежа. Магазин возвращает в PGA ответ, содержащий результат регистрации платежа (RPRes).
  16. PGA передает результат авторизации на платежную страницу.
  17. Платежная страница отображает клиенту результат платежа. Операция завершена.

## Сценарий выполнение платежа, инициированного клиентом, без повторного ввода параметров карты

Сценарий выполнения платежа без повторного ввода реквизитов карты клиентом представлен на Рисунок 3.

Данный сценарий позволяет выполнить платеж без повторного ввода реквизитов карты клиентом. Клиент вводит только CVV2/CVC2. Карта должна быть зарегистрированная в PGA или ранее должна быть выполнена 3DS-транзакция в адрес данного магазина.



Рисунок 3 Сценарий выполнение платежа, инициированного клиентом, без повторного ввода параметров карты

Прямой поток событий сценария выполнение платежа, инициированного клиентом, без повторного ввода параметров карты, выглядит следующим образом:

* 1. Клиент формирует заказ на сайте интернет-магазина и инициирует оплату сформированного заказа.
  2. Клиент перенаправляется на адрес платежной страницы, содержащий параметры операции и идентификатор магазина.
  3. Платежная страница формирует StartPaymentReq.
  4. PGA передает в магазин запрос проверки возможности проведения платежа (CPAReq).
  5. Магазин возвращает в PGA ответ о возможности проведения операции (CPARes). Ответ содержит идентификатор ранее зарегистрированной карты с указанием типа операции (card.present=Y) и дополнительные атрибуты.
  6. PGA определяет, зарегистрирована ли карта с полученным идентификатором для данного магазина. Успешно.
  7. PGA отправляет в платежную страницу сообщение с информацией о платеже для подтверждения пользователем.
  8. Клиенту отображается соответствующая платежная страница с маскированным PAN.
  9. Клиент вводит значение кода CVV2/CVC2. Параметры карты передаются в PGA.
  10. PGA взаимодействует с Directory Server для проверки участия банковской карты в программе 3-D Secure. Успешно.
  11. PGA запрашивает аутентификацию клиента.
  12. Клиент перенаправляется на аутентификацию в ACS Эмитента.
  13. Клиент проходит 3DS-аутентификацию (PAReq).
  14. PGA через браузер получает от сервера аутентификации банка-эмитента сообщение с результатом аутентификации (PARes).
  15. PGA направляет в ПЦ банка-эквайера запрос авторизации на сумму покупки. ПЦ банка-эквайера возвращает результат авторизации в PGA.
  16. PGA направляет в магазин запрос на регистрацию платежа (RPReq). Магазин возвращает в PGA ответ, содержащий результат регистрации платежа (RPRes).
  17. PGA передает результат авторизации на платежную страницу.
  18. Платежная страница отображает клиенту результат платежа. Операция завершена.

**Альтернативный поток событий:**

* 1. Шаг 1.4 прямого потока событий. Магазин возвращает в PGA ответ о возможности проведения операции (CPARes). Ответ содержит идентификатор ранее проведенной 3DS-транзакции с указанием типа операции (card.present=Y) и дополнительные атрибуты.
  2. PGA по полученному идентификатору выполняет поиск транзакции для данного магазина. Успешно.
  3. Управление передается на шаг 1.6 прямого потока событий.

## Сценарий выполнения рекуррентного платежа, инициированного магазином

Сценарий платежа, инициированный магазином, без участия клиента и повторного ввода параметров карты, представлен на Рисунок 4.

Данный сценарий позволяет выполнять платеж по инициативе магазина, без участия клиента и, соответственно, без повторного ввода параметров карты.



Рисунок 4 Сценарий выполнения рекуррентного платежа, инициированного магазином

Прямой поток событий сценария платежа, инициированный магазином, без участия Клиента и повторного ввода параметров карты, выглядит следующим образом:

* 1. Магазин, при наступлении определенного клиентом события, направляет в PGA запрос, инициирующий платеж.
  2. PGA передает в магазин запрос проверки возможности проведения платежа (CPAReq).
  3. Магазин возвращает в PGA ответ о возможности проведения операции (CPARes). Ответ содержит идентификатор ранее зарегистрированной карты с указанием типа операции (card.present=N) и дополнительные атрибуты.
  4. PGA выполняет проверку возможности проведения рекуррентных платежей для данного магазина. Проверка выполняется успешно (recurrentAllowed=Y).
  5. PGA по идентификатору карты определяет, зарегистрирована ли карта для проведения рекуррентных платежей в адрес данного магазина. Успешно.
  6. PGA формирует и направляет в ПЦ банка-эквайера авторизационный запрос на сумму платежа с признаком реккурента. ПЦ банка-эквайера возвращает результат рекуррентной авторизации в PGA.
  7. PGA направляет в магазин запрос на регистрацию платежа (RPReq) с указанием соответствующего кода завершения. Магазин возвращает в PGA ответ, содержащий результат регистрации платежа (RPRes). Операция завершена.
  8. PGA возвращает в магазин ответ с результатом платежа. Операция завершена.

**Альтернативный поток событий:**

* 1. Шаг 1.2 прямого потока событий. Магазин возвращает в PGA ответ о возможности проведения операции (CPARes). Ответ содержит идентификатор ранее проведенной 3DS-транзакции с указанием типа операции (card.present=N) и дополнительные атрибуты.
  2. PGA выполняет проверку возможности проведения рекуррентных платежей для данного Магазина. Проверка выполняется успешно (recurrentAllowed=Y).
  3. PGA по полученному идентификатору выполняет поиск транзакции для данного магазина. Успешно.
  4. Управление передается на шаг 1.5 прямого потока событий.

## Сценарий проведения платежа Газпромпей с определения средства платежа на платежной странице

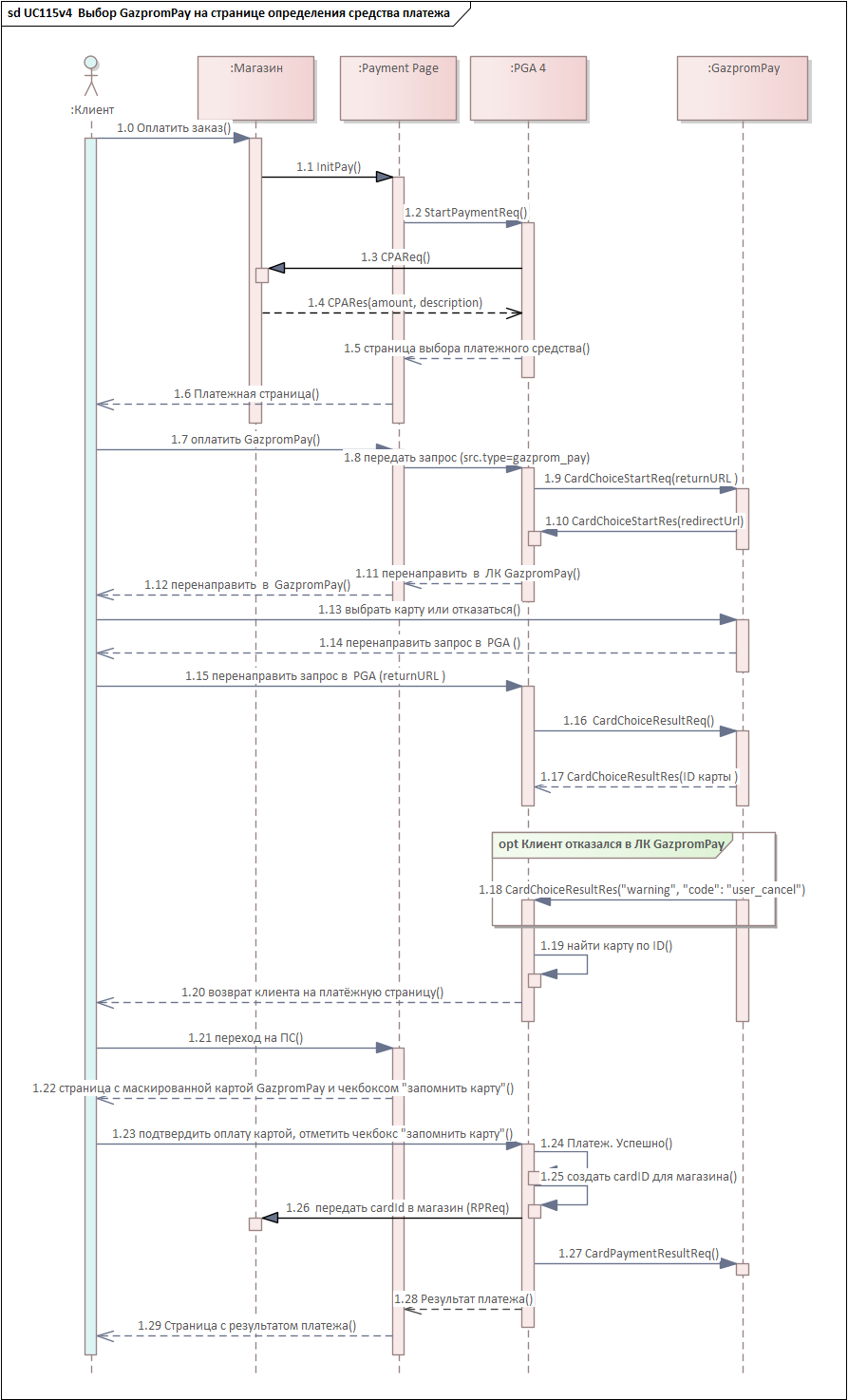


Рисунок 5 Сценарий проведения платежа Газпромпей с определения средства платежа на платежной странице

Описание шагов сценария:

* 1. Клиент инициирует оплату.
  2. Магазин перенаправляет запрос на платежную страницу.
  3. Платежная страница запрашивает старт платежа в PGA.
  4. PGA запрашивает проверку возможности проведения платежа в ТСП.
  5. ТСП отправляет ответ о возможности проведения платежа в PGA. Согласие магазина на сохранение карты, получаемое от ТСП в этом сценарии игнорируется.
  6. PGA отправляет запрос на показ Клиенту страницы для выбора платежного средства, в сообщении указан признак, что для магазина разрешена работа с GazpromPay.
  7. На платежной странице Клиенту отображается страница выбора средства с кнопкой оплаты с помощью GazpromPay.
  8. Клиент выбирает способ оплаты "GazpromPay".
  9. От платежной страницы в PGA отправляется запрос, в котором подтверждается согласие Клиента с условиями платежа. В запросе содержится поле src.type=gazprom\_pay.
  10. PGA выполняет запрос CardChoiceStartReq в GazpromPay, где передаёт адрес возврата в PGA.
  11. PGA получает от GazpromPay url переадресации в ответе CardChoiceStartRes.
  12. PGA перенаправляет Клиента в GazpromPay
  13. Клиент через платежную страницу перенаправляется в GazpromPay.
  14. Клиент выбирает карту.
  15. После выбора карты GazpromPay отправляет запрос в PGA (возврат по returnURL)
  16. Перенаправление запроса через браузер Клиента в PGA
  17. PGA запрашивает cardId в GazpromPay в сообщении CardChoiceResultReq.
  18. GazpromPay в ответе CardChoiceResultRes передает cardId в PGA.
  19. Опционально. Клиент отказался от выбора карты в личном кабинете GazpromPay.
  20. PGA находит карту в БД по cardId.
  21. PGA выполняет возврат Клиента на ПС.
  22. Клиент переходит на ПС.
  23. Клиенту отображается страница подтверждения карты, полученной от GazpromPay, на которой карта отображается в маскированном виде. На странице присутствует чекбокс «Запомнить мою карту», согласно REQ455v.3 UI.
  24. Клиент подтверждает оплату с помощью GazpromPay. Выбирает чекбокс «Запомнить мою карту».
  25. PGA выполняет платеж. Успешно.
  26. PGA формирует идентификатор карты для магазина. Карта сохраняется.
  27. PGA отправляет RPReq с новым идентификатором карты в магазин, где идентификатор карты будет сохранен.
  28. PGA отправляет сообщение CardPaymentResultReq в GazpromPay
  29. PGA отправляет сообщение с результатом платежа на ПС.
  30. ПС с результатом платежа отображается Клиенту.

Опционально, если при взаимодействии PGA с GazpromPay получено предупреждение "Отказ пользователя от оплаты" ("warning", "code": "user\_cancel") необходимо завершать платеж как неуспешный.

## Сценарий проведения платежа Газпромпей с определения средства платежа на стороне ТСП

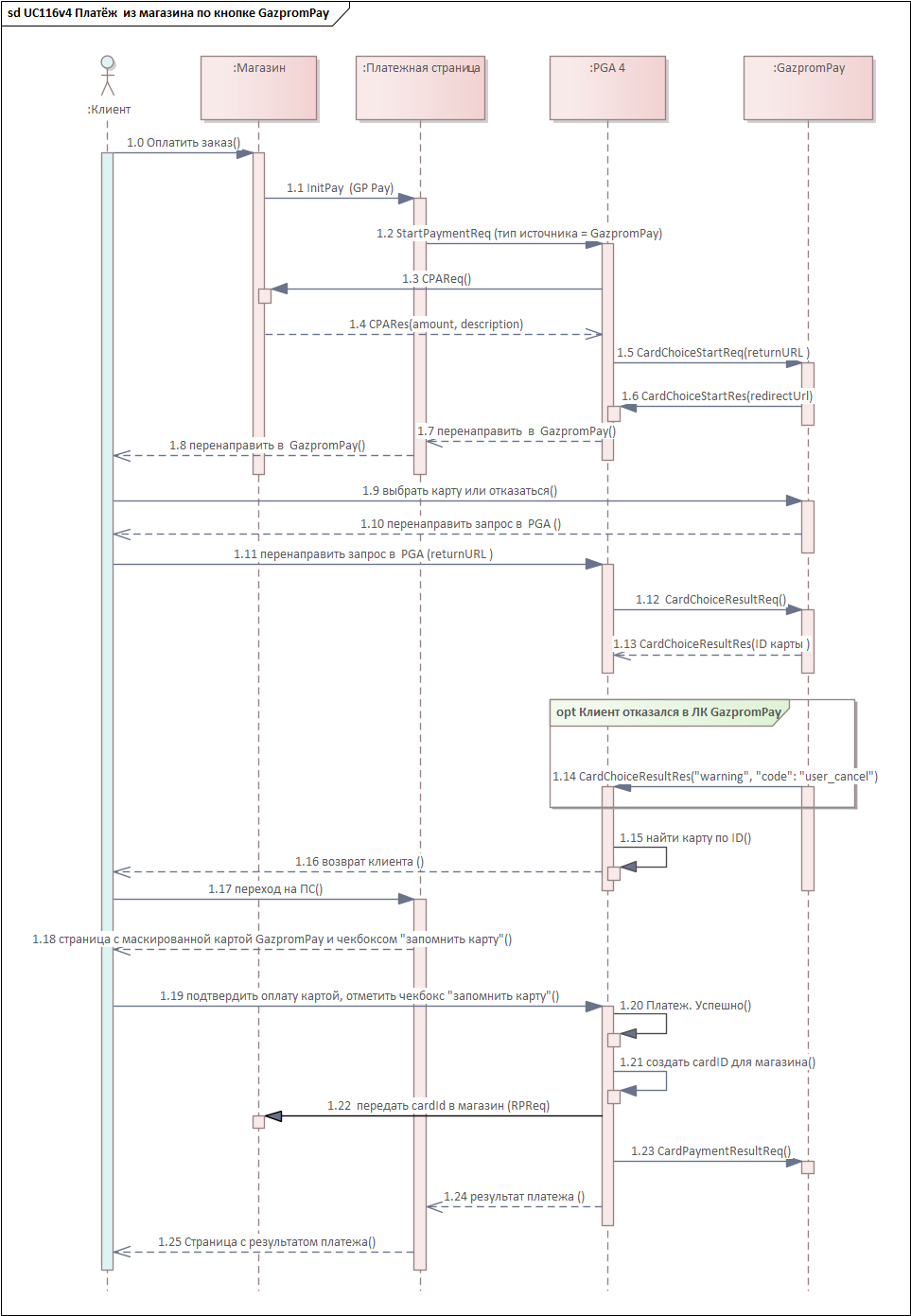


Рисунок 6 Сценарий проведения платежа Газпромпей с определения средства платежа на платежной странице

Прямой поток событий сценария взаимодействия:

* 1. Клиент инициирует оплату на сайте магазина.
  2. Магазин перенаправляет запрос на платежную страницу (передается тип источника = GazpromPay).
  3. Платежная страница отправляет в PGA запрос StartPaymentReq, с указанием типа источника GazpromPay.
  4. PGA запрашивает проверку возможности проведения платежа в ТСП.
  5. ТСП отправляет ответ о возможности проведения платежа в PGA. Согласие магазина на сохранение карты, получаемое от ТСП, в этом сценарии игнорируется.
  6. PGA выполняет запрос CardChoiceStartReq в GazpromPay, где передаёт адрес возврата в PGA.
  7. PGA получает от GazpromPay url переадресации в ответе CardChoiceStartRes.
  8. 1.8 PGA выполняет перенаправление в GazpromPay. через платежную страницу.
  9. Клиент выбирает карту.
  10. 1.11 После выбора карты GazpromPay отправляет запрос в PGA (возврат в PGA по returnURL)
  11. PGA запрашивает cardId в GazpromPay в сообщении CardChoiceResultReq.
  12. GazpromPay в ответе CardChoiceResultRes передает cardId в PGA.
  13. Опционально. Клиент отказался от выбора карты в ЛК GazpromPay.
  14. PGA находит карту в БД по cardId.
  15. – 1.17 PGA отображает Клиенту платежную страницу.
  16. Клиенту отображается страница подтверждения карты, полученной от GazpromPay, на которой карта отображается в маскированном виде. На странице присутствует чекбокс «Запомнить мою карту», согласно REQ454v.3 UI.
  17. Клиент подтверждает оплату с помощью GazpromPay. Выбирает чекбокс «Запомнить мою карту».
  18. PGA выполняет платеж. Успешно.
  19. PGA формирует идентификатор карты для магазина. Карта сохраняется.
  20. PGA отправляет RPReq с новым идентификатором карты в магазин, где идентификатор карты будет сохранен.
  21. PGA отправляет сообщение CardPaymentResultReq в GazpromPay
  22. PGA отправляет сообщение с результатом платежа на ПС.
  23. ПС с результатом платежа отображается Клиенту.

Альтернативно, если при взаимодействии PGA с GazpromPay получена ошибка, то необходимо завершать платеж как неуспешный.

Альтернативно, шаг 1.20 если платеж не успешный, то идентификатор карты для магазина не создается.

Альтернативно, если клиент не выбрал чекбокс “Запомнить мою карту» на странице с маскированной картой Gazprompay, то идентификатор карты для магазина не создается.

Создание идентификатора карты (шаг 1.21) и передача его в сообщении RPReq возможно только для успешных платежей по GazpromPay.

## Сценарий проведения In-App платежа Mir Pay для МП ТСП с отменой



Рисунок 7 Диаграмма последовательности для In-App платежа

**Сценарий платежа**

(платежные страницы и PGA работают как шлюз для инициации In App-операции):

1. Клиент в мобильном приложении ТСП нажимает кнопку для инициации заказа.
2. Альтернативно: ТСП перенаправляет Клиента на платежную страницу PGA отправкой запроса StartPaymentReq с необязательным полем openApiMirPaySupported=true. Платежная страница поддерживает новый функционал «Mir Pay». (Наименование поля может измениться во время разработки).
3. Альтернативно: ТСП перенаправляет Клиента на платежную страницу PGA отправкой запроса StartPaymentReq без поля openApiMirPaySupported, либо его значение равно false. Платежная страница поддерживает текущий функционал «Mir Pay» до версии PGA 4.2.83 включительно. (Наименование поля может измениться во время разработки).
4. Платежная страница по OpenAPI отправляет в PGA запрос на старт платежа.
5. PGA выполняет проверку доступности платежа в магазине запросом CPAReq.
6. ТСП подтверждает платеж сообщением CPARes.
7. PGA отвечает на запрос на старт платежа (шаг.1.3) статусом «offer». Сервер PGA передает JWT (merchantToken) и JWT1 (paymentToken) ( REQ770). JWT используется для активации приложения Mir Pay (merchantToken п.3.1) и JWT1 для инициации In-Application операции (paymentToken п.3.2).
8. Опционально:Платежная страница показывает Клиенту WEB-форму для ввода реквизитов банковской карты и кнопку «Mir Pay». Если платежная страница запущена в браузере на desktop, то кнопка не отображается.

Если в сообщении на старт платежа указан src.type=mir\_pay, то шаг в последовательности отсутствует.

1. Опционально:Клиент нажимает на кнопку «Mir Pay».

Если в сообщении на старт платежа указан src.type=mir\_pay, то шаг в последовательности отсутствует.

1. Альтернативно: если включена настройка «Отправить Deep Link» платежная страница отправляет Deep Link запрос на запуск приложения «Mir Pay».
2. Альтернативно: если включена настройка «Отправить Universal Link» платежная страница отправляет Universal Link запрос на запуск приложения «Mir Pay».
3. Альтернативно: Платежная страница отправляет запрос «mirpay/prepare» в PGA . Признак того, что платежные страницы поддерживают новый протокол «Mir Pay» (1.11 и 1.13 необходимы для того, чтобы платежная страница могла сообщить PGA, что надо ожидать JWE).
4. Альтернативно: Платежная страница отправляет запрос «mirpay accept» в PGA. Признак того, что платежные страницы поддерживают старый протокол «Mir Pay». Ответ /api/ "state":"result"
5. Ответ PGA статусом «offer» (mirPayInProgress=true). Только для нового протокола «Mir Pay».
6. Приложение «Mir Pay» отправляет в PGA запрос  
   GET /api/v4/${portal}/payment/${token}/mirpay/jwks на получение конечного сертификат ТСП.
7. PGA передает цепочку сертификатов JWKS в «Mir Pay».
8. PGA получив запрос п. **Error! Reference source not found.** понимает, что транзакция идет по потоку MirPay (In App-операция) и ожидает JWE.
9. Приложение «Mir Pay» запускается и проверяет подпись PGA на JWT1 в НСПК МИР. Альтернативно: если «Mir Pay» не установлен на устройстве, то Клиент увидит модальное окно с возможностью отмены In App-операции. Выполнение альтернативного сценария далее на сиквенс-диаграмме не показано. (См.Рис.3).
10. Альтернативно: Проверка «ТСП является участником In-App» положительная. Поля участника заполнены, сертификаты для «Mir Pay» берутся из схемы ТСП.

Альтернативно: Проверка «ТСП является участником In-App» отрицательная. Поля участника не заполнены, то данные для «Mir Pay» берутся из протокола «Mir Pay».

1. Приложение «Mir Pay» предлагает Клиенту интерфейс для выбора банковской карты.
2. Клиент выбирает карту, переход на шаг **Error! Reference source not found.** и далее.
3. Альтернативно: Клиент отказался выбрать карту, нажал «Отмена».
4. Альтернативно: Клиент самостоятельно вернулся на платежную страницу, завершил платеж или выбрал другой способ оплаты, не «Mir Pay». Выполнение альтернативного сценария далее на сиквенс-диаграмме не показано.
5. Приложение «Mir Pay» дообогащает JWT1 до JWT2.
6. Приложение «Mir Pay» отправляет запрос на подпись и шифрование JWT2 в НСПК МИР.
7. JWT2 подписывается с помощью [SK.NSPK.1] и затем шифруется на открытом ключе [PK.APPREG.1] – получаем JWE.
8. JWE передается в приложение «Mir Pay».
9. Приложение «Mir Pay» по протоколу «MirPay» API отправляет JWE в PGA.   
   Запрос: PUT /api/v4/${portal}/payment/${token}/mirpay/accept
10. Альтернативно: «JWE обработан успешно», переход на п. 1.30.
11. Альтернативно: «error: " 400 Bad Request"».
12. PGA расшифровывает JWE с помощью [SK.APPREG.1] и проверяет подпись с помощью [PK.NSPK.1].
13. Альтернативно: JWE не корректный. Запрос статуса платежа платежной страницей.
14. PGA отправляет ответ с результатом платежа.Ответ – error: "CARD\_TOKEN\_UNRESOLVED". Возврат Клиента к возможности выбора другого способа оплаты.
15. Альтернативно: JWE корректный. PGA отправляет авторизационной запрос в ПЦ.
16. ПЦ отправляет в PGA ответ. Успешно.
17. Запрос на регистрацию платежа от PGA в ТСП.
18. Ответ от ТСП.
19. Запрос статуса платежа платежной страницей.
20. Альтернативно: Ответ с результатом платежа.
21. Платежная страница показывает Клиенту результат платежа.
22. пункты 1.10, **Error! Reference source not found.**, **Error! Reference source not found.**, **Error! Reference source not found.** и **Error! Reference source not found.** добавлены для понимания.
23. в JWT1 в поле rurl указывается URL запроса по пункт **Error! Reference source not found.**.

## Сценарий проведения In-App платежа Mir Pay для ТСП с библиотекой MirPaySDK.aar с отменой

****

Рисунок 3 Диаграмма последовательности для In-App платежа в МП ТСП с библиотекой MirPaySDK.aar

**Сценарий платежа**

(PGA работает как шлюз для инициации In App-операции):

1. ТСПApp по OpenAPI отправляет в PGA запрос на старт платежа.
2. PGA выполняет проверку доступности платежа в ТСП запросом CPAReq.
3. ТСП подтверждает платеж сообщением CPARes.
4. PGA отвечает на запрос на старт платежа статусом «offer». Сервер PGA передает JWT (merchantToken) и JWT1 (paymentToken) ( REQ770). JWT используется для активации приложения Mir Pay и JWT1 для инициации In-Application операции.
5. ТСПApp проверяет наличие установленного приложения Mir Pay на устройстве пользователя с помощью метода MirApp.isMirPayInstalled.
6. ТСПApp отправляет запрос «mirpay/prepare» в PGA . Признак того, что PGA надо ожидать JWE).
7. Ответ PGA статусом «offer» (mirPayInProgress=true).
8. ТСПApp активирует приложение Mir Pay на устройстве с помощью метода MirPayPaymentClientFactory.create. Формируется JWT.
9. PGA получив запрос понимает, что транзакция идет по потоку MirPay (In App-операция) и ожидает JWE.
10. ТСПApp инициирует сценарий формирования In-Application операции с помощью метода IMirPayPaymentClient.createPaymentIntent. Формируется JWT1.
11. ТСПApp получает данные In-Application с помощью метода MirPayPaymentResultExtractor.extract.
12. Приложение Mir Pay отправляет в ТСПApp paymentResult.
13. ТСПApp отправляет JWE в PGA.   
    Запрос: PUT /api/v4/${portal}/payment/${token}/mirpay/accept
14. ТСПApp завершает взаимодействие с приложением Mir Pay с помощью метода IMirPayPaymentClient.disconnect.
15. PGA отправляет ответ ТСПApp.
16. PGA расшифровывает JWE с помощью [SK.APPREG.1] и проверяет подпись с помощью [PK.NSPK.1].
17. JWE корректный. PGA отправляет авторизационной запрос в ПЦ.
18. ПЦ отправляет в PGA ответ. Успешно.
19. PGA делает запрос на регистрацию платежа RPReq в ТСП.
20. ТСП отправляет в PGA ответ RPRes.

## Сценарий проведения AFT операций с вводом данных карты на платежной странице

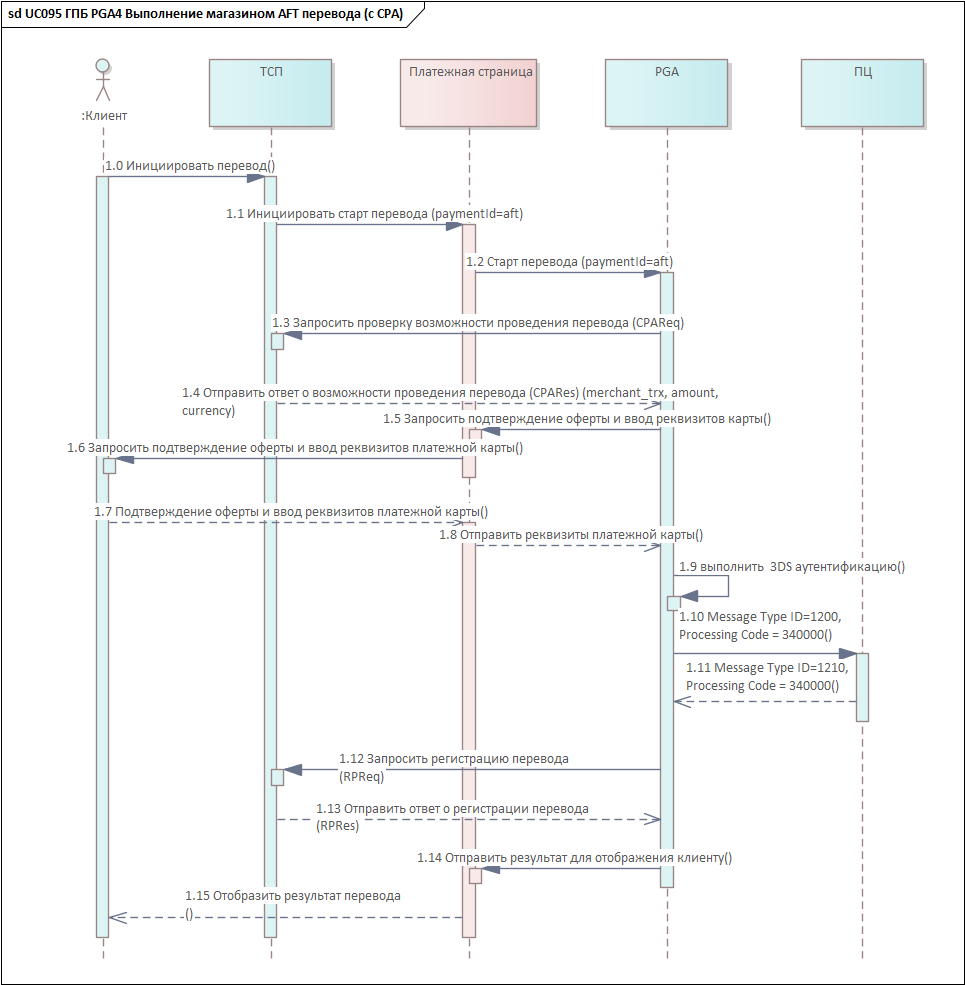
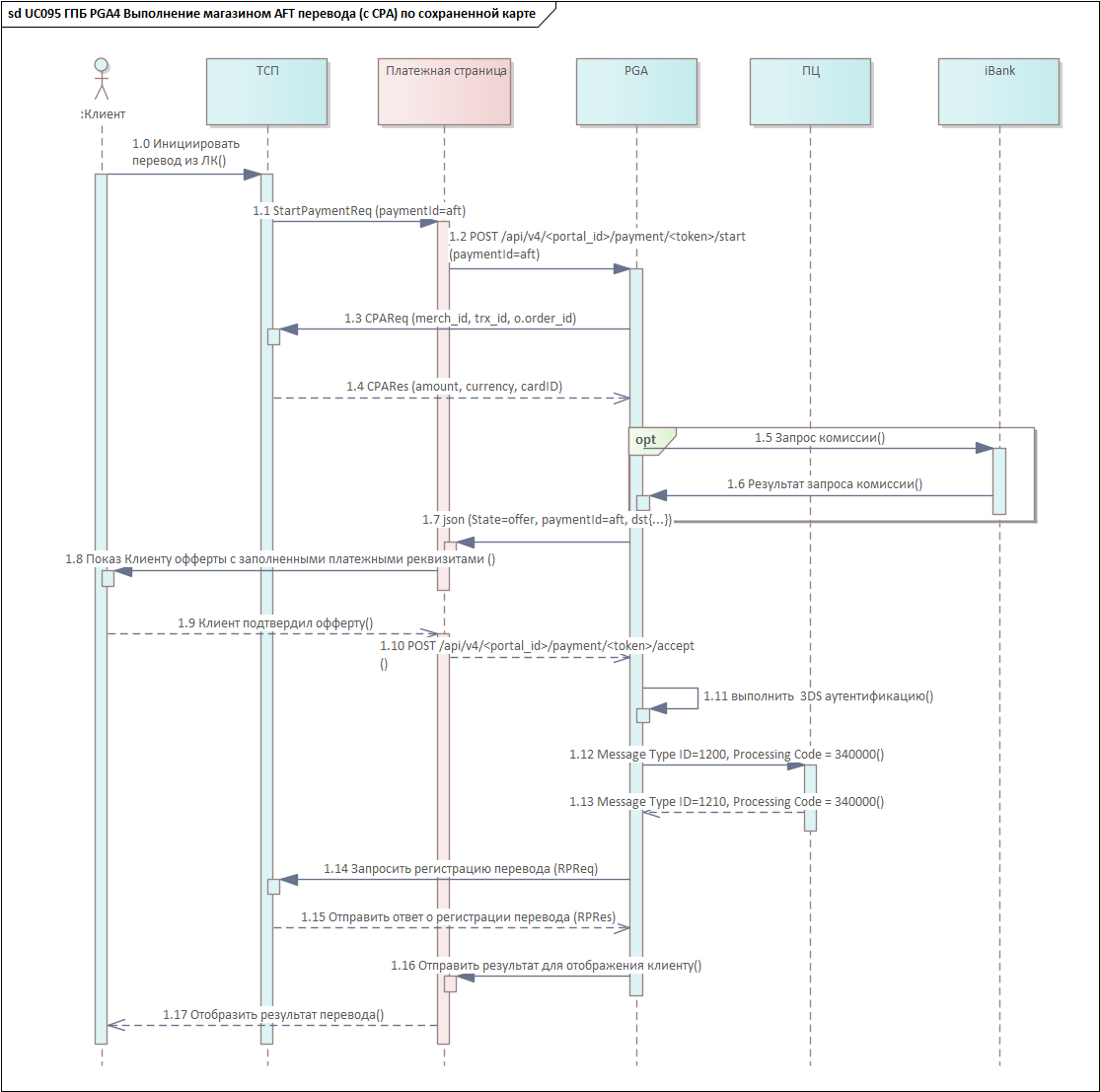


Рисунок 8 Сценарий проведения AFT операций с вводом данных карты на платежной странице

Прямой поток событий сценария проведения AFTопераций с вводом данных карты на платежной странице:

* 1. Клиент на WEB-интерфейсе ТСП нажимает кнопку для инициации перевода.
  2. ТСП инициирует начало проведения перевода, на старте передается признак paymentId=”aft” через платежную страницу PGA.
  3. Платежная страница PGA отправляет запрос на старт перевода (по протоколу OpenAPI).
  4. PGA запрашивает проверку возможности проведения перевода в ТСП.
  5. ТСП отправляет ответ о возможности проведения перевода в PGA, передаются параметры amount, currency.
  6. PGA отправляет запрос на подтверждение оферты и ввод реквизитов карты (ответ на старт перевода, п. 1.2).
  7. Платежная страница PGA показывает Клиенту оферту для ввода реквизитов карты.
  8. Клиент вводит реквизиты платежной карты и подтверждает оферту на платежной странице.
  9. Платежная страница передает реквизиты карты в PGA.
  10. PGA выполняет 3DS аутентификацию для AFT операции.
  11. PGA отправляет авторизационный запрос типа AFT.
  12. ПЦ отправляет в PGA результат выполнения авторизационного запроса.
  13. PGA отправляет запрос в ТСП на регистрацию перевода.
  14. ТСП отправляет в PGA ответ о регистрации перевода.
  15. PGA отправляет на платежную страницу результат операции для отображения Клиенту.
  16. Платежная страница отображает результат операции Клиенту.

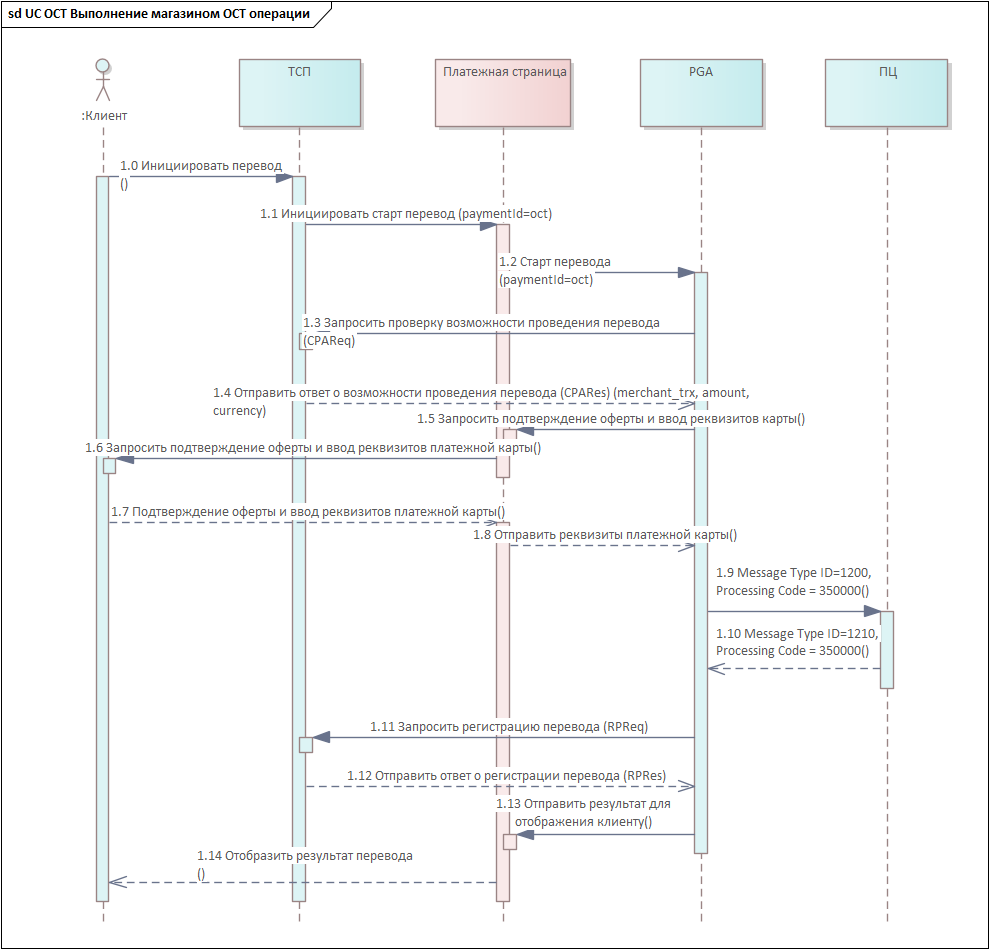
## Сценарий проведения AFT операций по ранее сохраненной карте

**Рисунок 9 Сценарий проведения AFT операций по ранее сохраненной карте**

Прямой поток событий сценария проведения магазином AFT операций в ПО по ранее сохраненной карте:

* 1. Клиент инициирует перевод в ЛК магазина.
  2. Магазин перенаправляет запрос на платежную страницу.
  3. Платежная страница запрашивает старт перевода в ПО, paymentId=”aft”.
  4. ПО запрашивает проверку возможности проведения перевода в магазине (CPAReq).
  5. ТСП отправляет ответ о возможности проведения перевода в ПО, передавая параметры amount, currency, transaction\_type= AFT, card\_id (CPARes).
  6. Опционально ПО запрашивает комиссию.
  7. Опционально получен ответ на запрос комиссии.
  8. ПО запрашивает подтверждение оферты и передает платежные данные (ответ на сообщение 1.2).
  9. Платежная страница отображает клиенту оферту с платежными данными (маскированный PAN).
  10. Клиент подтверждает оферту.
  11. Платежная страница отправляет в ПО запрос «Подтверждение платежа».
  12. ПО выполняет 3DS аутентификацию для AFT операции.
  13. ПО отправляет авторизационный запрос.
  14. ПЦ отправляет в ПО результат выполнения авторизационного запроса.
  15. ПО отправляет запрос в ТСП на регистрацию перевода.
  16. ТСП отправляет в ПО ответ о регистрации перевода.
  17. ПО отправляет результат операции для отображения Клиенту.
  18. Платежная страница отображает клиенту результат перевода.

## Сценарий проведения OCT операций с вводом данных карты на платежной странице

**Рисунок 10 Сценарий проведения OCT операций с вводом данных карты на платежной странице**

Прямой поток событий сценария проведения OCT операций с вводом данных карты на платежной странице:

* 1. Клиент на WEB-интерфейсе ТСП нажимает кнопку для инициации перевода.
  2. ТСП инициирует начало проведения перевода, на старте перевода передается признак paymentId=”oct” через платежную страницу PGA.
  3. Платежная страница PGA отправляет запрос на старт перевода (по протоколу OpenAPI).
  4. PGA запрашивает проверку возможности проведения перевода в ТСП.
  5. ТСП отправляет ответ о возможности проведения перевода в PGA, передаются параметры amount, currency.
  6. PGA отправляет запрос на подтверждение оферты и ввод реквизитов карты (ответ на старт перевода, п. 1.2).
  7. Платежная страница PGA показывает Клиенту оферту для ввода реквизитов карты.
  8. Клиент вводит реквизиты платежной карты и подтверждает оферту на платежной странице.
  9. Платежная страница передает реквизиты карты в PGA
  10. PGA отправляет в ПЦ авторизационный запрос.
  11. ПЦ отправляет в PGA результат выполнения авторизационного запроса.
  12. PGA отправляет запрос в ТСП на регистрацию перевода.
  13. ТСП отправляет в PGA ответ о регистрации перевода.
  14. PGA отправляет на платежную страницу результат операции для отображения Клиенту.
  15. Платежная страница отображает результат операции Клиенту.

## Сценарий проведения OCT операций по сохраненной карте

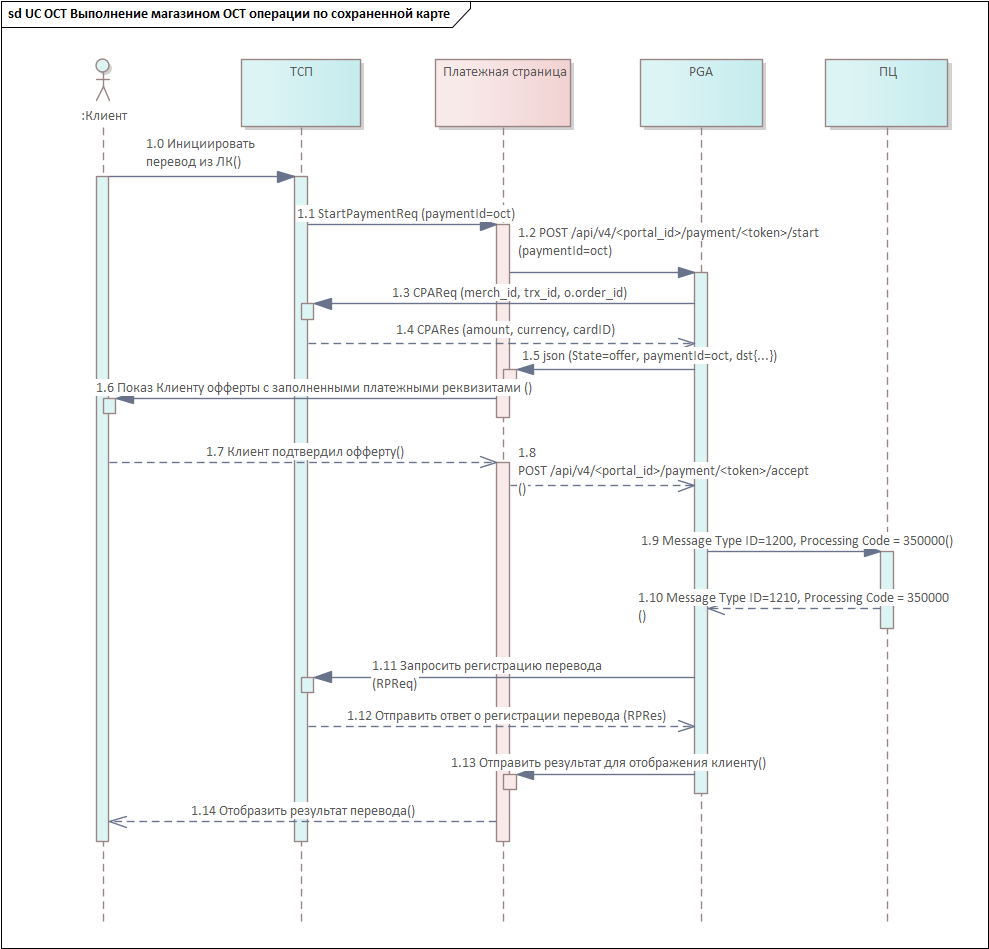
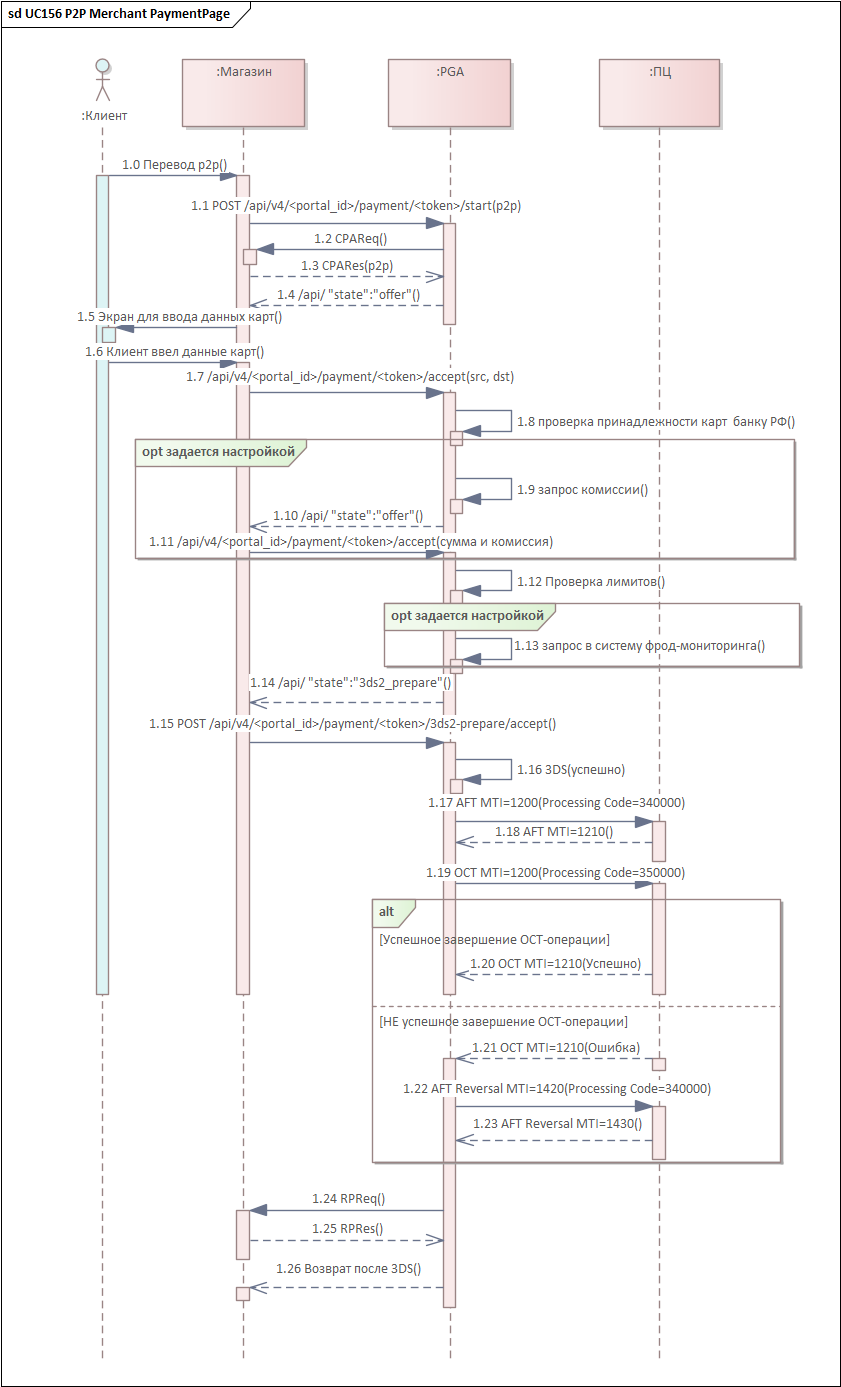


Рисунок 11 Сценарий проведения OCT операций по сохраненной карте

Прямой поток событий сценария проведения магазином OCT операций в ПО по сохраненной карте:

* 1. Клиент инициирует перевод в ЛК магазина.
  2. Магазин перенаправляет запрос старта перевода на платежную страницу.
  3. Платежная страница запрашивает старт перевода в ПО, paymentId=”oct”.
  4. ПО запрашивает проверку возможности проведения перевода в ТСП (merch\_id, trx\_id, o.order\_id).
  5. ТСП отправляет ответ о возможности проведения перевода в ПО, передавая параметры: amount, currency, cardID.
  6. ПО запрашивает подтверждение оферты и передает платежные данные (ответ на сообщение 1.2).
  7. Платежная страница отображает клиенту оферту с платежными данными (маскированный PAN).
  8. Клиент подтверждает оферту.
  9. Платежная страница отправляет в ПО запрос «Подтверждение платежа».
  10. ПО отправляет авторизационный запрос в ПЦ.
  11. ПЦ отправляет в ПО результат выполнения запроса.
  12. ПО отправляет запрос в ТСП на регистрацию перевода.
  13. ТСП отправляет в ПО ответ о регистрации перевода.
  14. ПО отправляет результат операции для отображения Клиенту.
  15. Платежная страница отображает клиенту результат перевода.

## Сценарий проведения P2P операций с вводом данных карты на платежной странице.



Прямой поток событий сценария проведения магазином Р2Р с вводом данных карты на платежной странице:

1. Клиент на витрине ТСП нажимает кнопку «Перевод p2p».
2. ТСП отправляет в PGA запрос на старт платежа с признаком «p2p».
3. PGA выполняет проверку доступности платежа в магазине запросом CPAReq.
4. ТСП подтверждает платеж сообщением CPARes (transaction-type=p2p).
5. PGA отвечает статусом offer.
6. Магазин на своей витрине запрашивает ввод данных карт: источника и получателя средств.
7. Клиент ввел данные карт и нажал кнопку «Перевести».
8. Магазин отправляет запрос accept с данными карт:
   * 1. Отправителя: src.type, src.pan, src.expiry, src.csc;
     2. Получателя: dst.type и dst.pan.
9. PGA выполняет проверку принадлежности src.pan и dst.pan российскому банку по ДНК (по полю «Страна»)
10. PGA выполняет запрос комиссии в сервис банка (если задано настройкой).
11. Опционально. PGA оповещает клиента о размере комиссии (второй offer).
12. Опционально. Клиент на витрине магазина согласился с суммой комиссии, магазин отправил второй запрос accept с подтверждением комиссии.
13. PGA выполняет проверку лимитов.
14. PGA выполняет запрос в систему фрод-мониторинга (если задано настройкой).
15. PGA отвечает статусом 3DS\_prepare.
16. Магазин выполняет запрос-подтверждение подготовки к 3DS
17. Выполняется 3DS. Успешно.
18. PGA выполняет AFT-запрос в БПЦ.
19. Ответ от БПЦ. Успешно.
20. PGA выполняет OCT-запрос в БПЦ.
21. Ответ от БПЦ. Успешно.
22. Альтернативный сценарий. Ответ от БПЦ. Ошибка.
23. Запрос на отмену AFT-операции.
24. Ответ от БПЦ на запрос отмены AFT-операции.
25. Запрос на регистрацию перевода в магазине.
26. Ответ магазина на запрос регистрации.
27. Возврат Клиента на витрину магазина после 3DS.

## Сценарий проведения P2P операций по сохраненной карте.

Прямой поток событий сценария проведения магазином Р2Р операций по сохраненной карте:

1. Клиент на витрине ТСП нажимает кнопку «Перевод p2p».
2. ТСП отправляет в PGA запрос на старт платежа с признаком «p2p».
3. PGA выполняет проверку доступности платежа в магазине запросом CPAReq.
4. ТСП подтверждает платеж сообщением CPARes (transaction-type=p2p, , card\_id).
5. PGA отвечает статусом offer.
6. Магазин на своей витрине запрашивает ввод данных карт: источника и получателя средств.
7. Клиент ввел данные карт и нажал кнопку «Перевести».
8. Магазин отправляет запрос accept с данными карт:
   * 1. Отправителя: src.csc;
     2. Получателя: dst.type и dst.pan.
9. PGA выполняет проверку принадлежности src.pan и dst.pan российскому банку по ДНК (по полю «Страна»)
10. PGA выполняет запрос комиссии в сервис банка (если задано настройкой).
11. Опционально. PGA оповещает клиента о размере комиссии (второй offer).
12. Опционально. Клиент на витрине магазина согласился с суммой комиссии, магазин отправил второй запрос accept с подтверждением комиссии.
13. PGA выполняет проверку лимитов.
14. PGA выполняет запрос в систему фрод-мониторинга (если задано настройкой).
15. PGA отвечает статусом 3DS\_prepare.
16. Магазин выполняет запрос-подтверждение подготовки к 3DS
17. Выполняется 3DS. Успешно.
18. PGA выполняет AFT-запрос в БПЦ.
19. Ответ от БПЦ. Успешно.
20. PGA выполняет OCT-запрос в БПЦ.
21. Ответ от БПЦ. Успешно.
22. Альтернативный сценарий. Ответ от БПЦ. Ошибка.
23. Запрос на отмену AFT-операции.
24. Ответ от БПЦ на запрос отмены AFT-операции.
25. Запрос на регистрацию перевода в магазине.
26. Ответ магазина на запрос регистрации.
27. Возврат Клиента на витрину магазина после 3DS.

# Сценарии регистрации карт

## Сценарий регистрации карты

Сценарий проведения регистрации карты клиента представлен на Рисунок 12.



Рисунок 12 Регистрация карты клиента

Прямой поток событий сценария выполнения платежа выглядит следующим образом:

* 1. Клиент инициирует процесс регистрации карты.
  2. Клиент перенаправляется на адрес страницы регистрации карты.
  3. Страница регистрации карты формирует StartCardRegReq и передает сообщение в PGA.
  4. PGA передает в магазин запрос проверки возможности проведения операции регистрации карты (CPAReq).
  5. Магазин возвращает в PGA ответ о возможности проведения операции регистрации карты (CPARes).
  6. PGA отправляет на страницу регистрации карты запрос о перенаправлении на страницу карты.
  7. Страница регистрации карты осуществляет перенаправление на страницу карты.
  8. Страница карты отображает клиенту запрос ввода реквизитов банковской карты.
  9. Клиент вводит реквизиты карты. В PGA направляются реквизиты карты.
  10. PGA взаимодействует с Directory Server для проверки участия банковской карты в программе 3-D Secure (VEReq).
  11. Directory Server отправляет в PGA успешный ответ об участии карты в программе 3-D Secure (VERes).
  12. PGA запрашивает аутентификацию клиента.
  13. Клиент перенаправляется на аутентификацию в ACS Эмитента.
  14. Клиент проходит 3DS-аутентификацию (PAReq).
  15. PGA через браузер получает от сервера аутентификации банка-эмитента сообщение с результатом аутентификации (PARes).
  16. PGA направляет в ПЦ банка-эквайера запрос проверочной авторизации. ПЦ банка-эквайера возвращает результат проверочной авторизации в PGA.
  17. PGA направляет в ПЦ банка-эквайера запрос проверочной рекуррентной авторизации. ПЦ банка-эквайера возвращает результат проверочной рекуррентной авторизации в PGA.
  18. PGA направляет в ПЦ банка-эквайера запрос отмены проверочной рекуррентной авторизации. ПЦ банка-эквайера возвращает результат отмены проверочной рекуррентной авторизации в PGA.
  19. PGA направляет в магазин запрос на регистрацию результата регистрации карты (RPReq).
  20. Магазин возвращает в PGA ответ, содержащий результат регистрации запроса (RPRes).
  21. PGA направляет в ПЦ банка-эквайера запрос отмены проверочной авторизации. ПЦ банка-эквайера возвращает результат отмены проверочной авторизации в PGA.
  22. PGA передает результат регистрации карты на страницу регистрации карты.
  23. Страница регистрации карты отображает клиенту результат регистрации карты. Операция завершена.

# Протоколы взаимодействия

## Константы

Для всех транзакций используется portal\_id :

* - тестовое контур 5BAE2814BB16ED0AA4A146AA4A4E168D,
* - промышленный контур 4246218C81AF2F5E6DF9742C494ADF22.

Параметр Host: <host> используется:

* - тестовый контур lt.pga.gazprombank.ru,
* - промышленный контур www.pga.gazprombank.ru

Адрес для возврата returnUrl = https://gazprombank.ru

Во всех примерах ниже подставлены значения для тестового контура Газпромбанка.

## Инициирование платежа

Для инициирования платежа магазин, после окончания формирования корзины заказа клиентом, выполняет переадресацию запроса в адрес PGA методом GET. Параметры запроса (Таблица 4) в адресной строке должны быть разделены символом «&».

Пример запроса инициирования платежа:

https://lt.pga.gazprombank.ru/pages/?lang\_code=RU&merch\_id=<merchant\_id>&back\_url\_s=<success\_url> &back\_url\_f=<failure\_url>&o.order\_id=<order\_id.value>

Параметры запроса инициирования платежа:

* <merchant\_id> - значение идентификатора магазина на тестовом контуре;
* <success\_url> - значение адреса, на который будет направлен пользователь в случае успешной оплаты и нажатия кнопки вернуться в магазин;
* <failure \_url> - значение адреса, на который будет направлен пользователь в случае неуспешной оплаты и нажатия кнопки вернуться в магазин;
* o.order\_id – о-параметр. Параметр o.order\_id является обязательным, т.к позволяет установить соответствие между заказом в системе магазина и транзакцией в системе банка. Набор остальных о-параметров является опциональным и определяется магазином, что позволяет реализовать другие необходимые элементы логики приложения магазина.
* <order\_id.value> - значение параметра o.order\_id.

Такой способ инициирования платежа используется в сценариях 4.1, 4.2, 4.4, 4.5, т.е. во всех сценариях, где используется платёжная страница PGA (см. Рисунок 13).

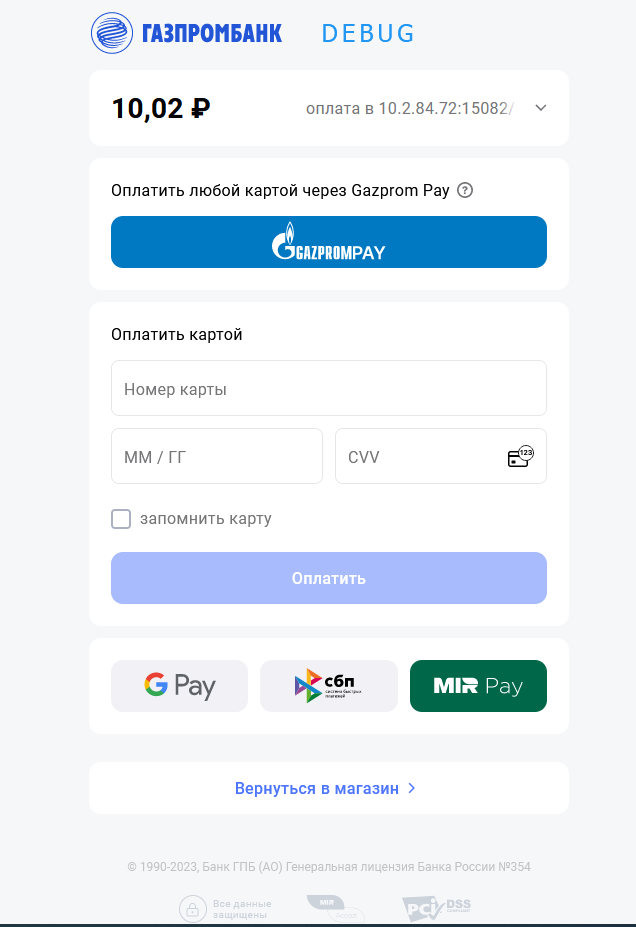


Рисунок 13 Платежная страница

* В случае необходимости магазин может отправить клиенту на email ссылку для инициирования платежа. Клиент может самостоятельно инициировать платеж, перейдя по ссылке из письма (отложенный платеж).

Таблица 4 Параметры запроса StartPaymentReq

| Параметр |  | Тип | Описание |
| --- | --- | --- | --- |
| lang\_code | О | string (2) | Язык платёжной страницы (двухсимвольный код в соответствии со стандартом ISO 639). |
| merch\_id | R | string (32) | Идентификатор магазина в PGA. |
| back\_url\_s | R | string (512) | Адрес, на который будет переадресован клиент после успешного завершения платежа. |
| back\_url\_f | R | string (512) | Адрес, на который будет переадресован клиент после неуспешного завершения платежа. |
| notify\_email | O | string (128) | Email для отправки уведомлений о результате операции клиенту.  Если email указан, то после завершения платежа выполняется настраиваемое количество попыток отправки результата операции на указанный адрес. При подключении магазина согласовываются настройки количества попыток отправки результата операции, интервал между ними и условие отправки (всегда, только успешные операции, только неуспешных операции, никогда). |
| preauthorize | O | string (5) | Флаг выполнения предварительной авторизации, т.е. блокировки средств. Возможные значения: true\false.  По умолчанию false. |
| o.\* | R | string (128) | Набор параметров покупки. Может содержать идентификатор транзакции, заказа и корзины в системе магазина. |
| paymentId | O | string (8) | Идентификатор платежа. Параметр используется для выбора одной из дополнительных платёжных схем. Если параметр не указан, будет выбрана основная платёжная схема.  Возможные значения: gambling, aft, oct, p2p, transp\_debt. |
| src.type | O | string (11) | Тип источника платежа.  Если параметр указан, по платёж будет начат с указанным типом источника платежа.  Возможные значения: gazprom\_pay, mir\_pay. |

Пример запроса инициирования платежа (StartPaymentReq):

https://lt.pga.gazprombank.ru/pages/?merch\_id=testMerchant&back\_url\_s=https://gazprombank.ru&back\_url\_f=https://gazprombank.ru&o.trx\_id=1

## Инициирование платежа по сохранённым реквизитам карты

Для сценариев выполнения по сохранённым реквизитам карты без участия клиента (4.3 и 5.3) инициирование платежа выполняется по протоколу OpenAPI ([2] Payguide Acquirer 4 Протокол OpenAPI Проведение платежей).

Магазин при этом сначала отправляет в PGA запрос на генерацию токена и получает в ответ токен. Потом магазин отправляет запрос на старт платежа и получает в ответ результат выполнения платежа или сообщение с промежуточным состоянием платежа.

Пример запроса на получение токена и ответа на него:

// Запрос магазином нового токена

POST/api/v4/5BAE2814BB16ED0AA4A146AA4A4E168D/token HTTP/1.1

Host: lt.pga.gazprombank.ru

Content-Type: application/x-www-form-urlencoded

Content-Length: 0

// Ответ с токеном

HTTP/1.1 200 OK

Content-Type: application/json

{«token»: «1234567890ABCDEF»}

Токен, полученный по результату платежа, необходимо указывать на старте платежа вместо парарметра <token>.

Пример запроса инициирования платежа по сохранённым реквизитам карты для платёжной схемы, которая работает без двухфазного взаимодействия с магазином:

// Запрос магазином старта платежа

POST /api/v4/5BAE2814BB16ED0AA4A146AA4A4E168D/payment/<token>/start HTTP/1.1

Content-Type: application/x-www-form-urlencoded

merchantId=MERCHANTIDENTIFICATOR

accountId=test\_account

src.type=card\_id  
src.cardId=ABCDEFGHIJKLMNOP  
params.order\_id=123  
amount=123456

currency=RUB  
recurrent=true  
state.redirect=post\_params  
state.in\_progress=no

returnUrl=https://gazprombank.ru

Пример запроса инициирования платежа по сохранённым реквизитам карты для платёжной схемы, которая работает с двухфазным взаимодействием с магазином (см. п. 6.4) :

// Запрос магазином старта платежа

POST /api/v4/5BAE2814BB16ED0AA4A146AA4A4E168D /payment/<token>/start HTTP/1.1

Content-Type: application/x-www-form-urlencoded

merchantId=MERCHANTIDENTIFICATOR

accountId=test\_account

params.order\_id=123  
state.redirect=post\_params  
state.in\_progress=no

returnUrl=https://gazprombank.ru

В этом примере магазин не указывает в запросе старта платежа данные карты, сумму, валюту и флаг рекуррента. Все эти параметры PGA получает от магазина в сообщении CPARes (см. п. 6.4.1).

Согласно спецификации EMVCo 3DS 2.2.0, для проведения платежей по сохранённым реквизитам карты, в AReq необходимо передавать следующие поля, предоставленные магазином:

* recurringExpiry (дата, после которой дальнейшие авторизации не должны выполняться, YYYYMMDD);
* recurringFrequency (минимальное количество дней между авторизациями).

Поля recurringExpiry и recurringFrequency явлюятся опциональными и передаются на старте платежа или в сообщении CPARes.

Пример с параметрами recurringExpiry и recurringFrequency на старте платежа:

// Запрос магазином старта платежа

POST /api/v4/5BAE2814BB16ED0AA4A146AA4A4E168D/payment/<token>/start HTTP/1.1

Content-Type: application/x-www-form-urlencoded

…

params.recurringExpiry=20250830

params.recurringFrequency=202

Пример с параметрами recurringExpiry и recurringFrequency в сообщении CPARes:

<payment-avail-response>

...

<merchant-trx>${merchantTrx}</merchant-trx>

<primaryTrxPcid>7EC2CD4BAD241028</primaryTrxPcid>

<order-params>

<param>

<name>recurringExpiry</name>

<value>20250830</value>

</param>

<param>

<name>recurringFrequency</name>

<value>31</value>

</param>

</order-params>

</payment-avail-response>

Начиная с версии 4.2.82, платеж по ранее сохраненным реквизитам карты разделяется на два подтипа: UCOF, MIT.

Для того, что бы сообщить PGA подтип платежа, магазин в сообщении CPARes для схемы с CPA или в старте платежа для схемы без CPA, должен указать параметр платежа с именем card\_on\_file и одним из двух значений (MIT или UCOF).

Пример сообщения CPARes для повторного платежа в схеме с CPA:

<?xml version=”1.0” encoding=”UTF-8”?>  
<payment-avail-response>  
 <result>  
 <code>1</code>  
 <desc>OK</desc>  
 </result>  
 <merchant-trx>merchant1</merchant-trx>

<card>  
 <id>123456789012</id>  
 </card>  
 <purchase>  
 <shortDesc>text</shortDesc>  
 <longDesc>text</longDesc>  
 <account-amount>  
 <id>1234567890</id>  
 <amount>1000</amount>  
 <currency>981</currency>  
 <exponent>2</exponent>  
 </account-amount>  
 </purchase>  
 <order-params>  
 <param>  
 <name>card\_on\_file</name>  
 <value>MIT</value>  
 </param>  
 </order-params>  
</payment-avail-response>

Пример старта платежа для повторного платежа в схеме без CPA:

merchantId= MERCHANTIDENTIFICATOR

description= Test Top-Up 79053324234

recurrent=true

src.type=card\_id

src.cardId= ABCDEFGHIJKLMNOP

params.card\_on\_file=UCOF

params.order\_id=ord-12345678

amount=55025

## Двухфазное взаимодействие с магазином

Двухфазное взаимодействие с магазином осуществляется по протоколу HTTPS c использованием метода GET в кодировке UTF-8. Ответы на запросы отправляются в формате XML.

**Взаимодействие** между сервисной платформой и магазином **синхронное.**

При возникновении ошибок транспортного уровня (коммуникационные ошибки, в том числе таймаут при ожидании ответа), отказа обработки запроса или отрицательного результата проверок, платеж не выполняется. Транзакция завершается с соответствующим кодом ошибки.

Протоколы двухфазного взаимодействия в PGA 4 и PGA 3 отличаются в части указания параметров карт. А также у них отличаются правила использования параметров p.rrn, p.authCode и p.isFullyAuthenticated запроса RPReq.

В настройках магазина в административной консоли есть возможность переключения на работу по протоколу PGA 3.

В описании запросов содержится признак обязательности каждого поля. Признаки обязательности полей приведены в Таблица 5.

Таблица 5 Признаки обязательности полей в сообщениях

| Обозначение параметра | Значение | Требования для отправителя | Требования для получателя |
| --- | --- | --- | --- |
| R | Обязательно. | Обязан включить поле в запрос. | Обязан проверить наличие и содержимое поля. |
| C | Обязательно при выполнении некоторого условия. | Обязан включить поле при выполнении соответствующего условия. | Обязан проверить наличие поля при выполнении условия. При наличии поля вне зависимости от условия обязан проверить содержимое. |
| O | Опционально. | Может включить поле в запрос. | Обязан проверить содержимое поля при его наличии. |

### Проверка доступности платежа в магазине

На первой фазе взаимодействия CheckPaymentAvailRequest (CPAReq) PGA с магазином производится проверка возможности проведения платежа, запрошенного клиентом. Запрос осуществляется методом GET и содержит параметры, указанные в Таблица 6.

Таблица 6 Параметры запроса CPAReq

| Параметр |  | Тип | Описание |
| --- | --- | --- | --- |
| merch\_id | R | string (32) | Идентификатор магазина в Сервисной платформе. |
| trx\_id | R | string (32) | Идентификатор транзакции в Сервисной платформе. |
| lang\_code | O | string (2) | Язык платёжной страницы (двухсимвольный код в соответствии со стандартом ISO 639). |
| o.\* | R | string (128) | Набор параметров покупки. |
| ts | R | string (17) | Дата выполнения запроса в формате «yyyyMMdd HH:mm:ss». |

Пример запроса проверки возможности проведения платежа:

https://merchant-website.ru/CheckPaymentAvail.php?merch\_id=IVACQ\_voyage&trx\_id=0MZN2KNHVPYQ4FAV&o.order\_id=114&lang\_code=RU&ts=20160511+17%3A02%3A11

Магазин формирует ответ CheckPaymentAvailResponse (CPARes) о возможности совершения покупки (в формате XML). Если платёж разрешён, то магазин должен указать параметры этого платежа: сумму, валюту, описание. При необходимости, магазин может указать также параметры карты, а также указать набор именованных параметров. Этот набор параметров PGA добавит в набор параметров платежа, которые были указаны в запросе старта платежа.

Параметры ответа приведены в Таблица 7. Для формирования ответа можно воспользоваться файлом в формате XSD (см. [Приложение 1](#_Приложение_1_Ответ)).

Таблица 7 Параметры ответа CPARes

| Параметр |  | Тип | Описание |
| --- | --- | --- | --- |
| result | R | tag | Описание результата проверки. |
| result.сode | R | integer (1) | Результат проверки возможности платежа:   * 1 - результат успешный; * 2 - результат неуспешный. |
| result.desc | O | string (125) | Описание результата проверки возможности платежа.  Рекомендуется заполнять идентификатором транзакции, заказа и корзины в системе магазина для разрешения возможных спорных ситуаций в случае их возникновения. |
| merchant-trx | O | string (50) | Идентификатор транзакции в системе магазина.  Рекомендуется использовать данный параметр для упрощения взаимодействия со службой поддержки. |
| submerchant | C | string(40) Допустимые символы: A-Za-z 0-9.,- | Имя подмагазина. Если параметр присутствует, то в качестве итогового имени магазина используется значение вида <имя магазина-посредника> + «\*» + <имя подмагазина> |
| purchase | С | tag | Описание покупки. Заполняется в случае, если result.сode=1. |
| purchase.shortDesc | O | string (30) | Краткое описание покупки. Если поле отсутствует, то в качестве краткого описания будет использовано полное описание, укороченное до 30-и символов. |
| purchase.longDesc | R | string (125) | Полное описание покупки. |
| purchase.account-amount | R | tag | Описание суммы покупки. |
| purchase.account-amount.id | O | string (32) | Идентификатор счета в системе магазина. Возвращается в запросе регистрации платежа (RPReq) в исходном виде. |
| purchase.account-amount.amount | R | integer (18) | Сумма платежа в минорных единицах. |
| purchase.  account-amount.  fee | O | integer (18) | Сумма комиссии в минорных единицах.  Эта комиссия носит информационный характер и не влияет на общую сумму платежа |
| purchase.account-amount.currency | R | integer (3) | Трехзначный цифровой код валюты (ISO 4217). |
| purchase.account-amount.exponent | R | integer (1) | Экспонента валюты платежа (ISO 4217). |
| order-params | O | tag | Описание дополнительных параметров платежа. |
| order-params.param | O | tag | Описание одного параметра платежа. |
| order-params.param.name | R | string (32) | Имя параметра платежа. |
| order-params.param.value | R | string (128) | Значение параметра платежа. |
| transaction-type | O | string (12) | Тип операции. Параметр может принимать следующие значения:   * CardRegister – операция регистрации карты; * Payment – платежная операция (по умолчанию); * AFT – операция AFT; * OCT – операция OCT. * P2P – операция P2P.   При обычном платеже с сохранением реквизитов карты тип операции должен отсутствовать или содержать значение Payment.  В транзакциях верификации карты тип операции также должен отсутствовать или содержать значение Payment. |
| card | O | tag | Описание ранее зарегистрированной карты или ранее проведённой транзакции |
| card.id | O | string (32) | Идентификатор банковской карты, которая должна использоваться в платеже, |
| card.trx-id | O | string (32) | Идентификатор транзакции, из которой необходимо брать параметры банковской карты. |
| card.present | O | string (1) | Идентификатор типа платежа. Параметр может принимать следующие значения:   * Y – обычный платеж без повторного ввода параметров карты (по умолчанию); * N – рекуррентный платеж. |
| card\_on\_file | C | string(4) | Может принимать значения:   * UCOF; * MIT (см. Таблицу «Термины и сокращения»). |
| mir-extension | С | tag | Передается в случае использования в качестве платежного средства карты ПС «МИР» |
| mir-extension.type | R | string (32) | Поддерживаются следующие типы идентификаторов получателя перевода:   * 3ds2.destCardNum; * 3ds2.destAcctNum; * 3ds2.destPhoneNum; * 3ds2.destWalletNum. |
| mir-extension.name | R | string (32) | Идентификатор получателя перевода |
| submerchant-data | С | tag | Параметры подмагазина (платежные схемы: «Интернет-эквайринг ГПБ», «AFT ГПБ», «P2P ГПБ», см. п. 6.4.1.1) |
| submerchant-data.city | R | string (..13) | Город (особенность обработки см. 6.4.1.1) |
| submerchant-data.country | R | string (3) | Страна (трехбуквенный код) |
| submerchant-data.id | R | string (15) | Идентификатор подмагазина |
| submerchant-data.name | R | string(..) | Наименование подмагазина. Рекомендации по формированию длины в п. 6.4.1.1 |
| submerchant-data.terminal-id | R | string (8) | Идентификатор терминала подмагазина |
| submerchant-data.mcc | R | string (4) | MCC |
| submerchant-data.inn | R | string (20) | ИНН |

Пример ответа на запрос возможности проведения платежа в случае, если магазин готов принять оплату:

<payment-avail-response>

<result>

<code>1</code>

<desc>Payment is available</desc>

</result>

<merchant-trx>Trx120</merchant-trx>

<purchase>

<shortDesc>Покупка книги.</shortDesc>

<longDesc>Путеводитель по Европе на сумму 1000 рублей.</longDesc>

<account-amount>

<id>A0CB3E876226313F6A7E2F220E7DFA1E</id>

<amount>100000</amount>

<fee>100</fee>

<currency>643</currency>

<exponent>2</exponent>

</account-amount>

</purchase>

</payment-avail-response>

Пример ответа на запрос возможности проведения платежа в случае, если магазин отказался принять оплату:

<payment-avail-response>

<result>

<code>2</code>

<desc>Unable to accept this payment</desc>

</result>

</payment-avail-response>

Пример ответа на запрос проверки доступности платежа в магазине:

<submerchantData>

<id></id>

<name></name>

<terminal-id></terminal-id>

<city></city>

<country></country>

<mcc></mcc>

<inn></inn>

</submerchantData>

### Передача параметров подмагазина

Для платежных схем «Интернет-эквайринг ГПБ» «AFT ГПБ» магазин-посредник (facilitator) должен передать информацию о подмагазине в сообщении CPARes (тэг <submerchant-data>, см. Таблица 7) .

Значение тэга <submerchant> в сообщении CPARes PGA игнорирует.

Если в настройках PGA установлен флаг "Магазин-посредник", а информация о подмагазине отсутствует в CPARes или соответствующие теги не заполнены, то транзакция завершиться ошибкой с кодом CPA\_FAILED.

Значение тега <submerchant-data.name> PGA использует для формирования составного поля {FacilitatorName}\*{submerchant-data.name} длиной 22 символа. Значение {FacilitatorName} берется из настроек в административной консоли PGA. При выборе значения тега <submerchant-data.name> необходимо учесть правила формирования составного поля {FacilitatorName}\*{ submerchant-data.name }, которые заключаются в следующем.

Префикс {FacilitatorName} не боле 11 символов:

• если больше – PGA обрезает его и добавляет \*{submerchant-data.name};

• если меньше добавляет \*{submerchant-data.name}.

Если общая длина поля {FacilitatorName}\*{submerchant-data.name} превышает 22 символа – PGA обрезает справа (обрезает постфикс {submerchant-data.name}). Если общая длина поля {FacilitatorName}\*{submerchant-data.name} меньше 22 символов - PGA заполняет справа пробелами.

Если длина тега <submerchant-data.city> больше 13 символов – PGA обрежет его справа, если меньше 13 символов - заполнит справа пробелами.

### Регистрация платежа в магазине

Для регистрации результата платежа PGA отправляет в магазин запрос RegisterPaymentRequest (RPReq). Запрос осуществляется методом GET и содержит параметры, указанные в Таблица 8.

Таблица 8 Параметры запроса RPReq

| Параметр |  | Тип | Описание |
| --- | --- | --- | --- |
| trx\_id | R | string (32) | Идентификатор транзакции в Сервисной платформе. |
| merch\_id | R | string (32) | Идентификатор магазина в Сервисной платформе. |
| merchant\_trx | O | string (50) | Идентификатор транзакции в системе магазина, полученный в CPARes (если магазин передал данный параметр). Рекомендуется использовать данный параметр для упрощения взаимодействия со службой поддержки. |
| result\_code | R | integer (1) | Результат платежа:   * 1 - результат успешный; * 2 - результат неуспешный. |
| ext\_result\_code | C | string(32) | Расширенный код завершения транзакции. Присутствует в случае, если result\_code=2.  Расширенные коды завершения платежа см. в Payguide\_Acquirer\_4\_Протокол\_OpenAPI\_Проведение\_платежей.docx (Приложение 2). |
| amount | R | integer (18) | Сумма платежа в минорных единицах. |
| account­\_id | O | string (125) | Идентификатор счета в системе магазина. Идентификатор получен от магазина в запросе CPARes и возвращается ему в исходном виде. |
| o.\* | R | string (512) | Набор параметров покупки. Передаются в том виде, как они были переданы при инициировании платежа и в сообщении CPARes. |
| p.rrn | O | string (40) | Идентификатор платежа в ПЦ банка-эквайера. Обязательно присутствует для успешной платёжной транзакции. В остальных случаях может отсутствовать.  В платежах СБП PGA не отправляет авторизационный запрос в процессинговый центр банка, поэтому в случае сбп параметр отсутствует. |
| p.authcode | O | string (20) | Авторизационный код, получен от ПЦ при успешно проведенной операции (ISO 8583 field 38). Обязательно присутствует для успешной платёжной транзакции. В остальных случаях может отсутствовать.  В платежах СБП PGA не отправляет авторизационный запрос в процессинговый центр банка, поэтому в случае сбп параметр отсутствует. |
| p.srcType | O | string (11) | Тип источника платежа. Возможные значения:   * CARD – незарегистрированная карта, * CARD\_ID – зарегистрированная карта, * GAZPROM\_PAY – платёж картой, сохранённой в кошельке Gazprom Pay, * SBP – платёж с помощью Системы Быстрых Переводов * MIR\_PAY – платеж Mir Pay |
| p.maskedPan | O | string (19) | Маскированный номер карты. |
| p.paymentSystem | O | string (10) | Платёжная система источника платежа. Возможные значения:   * VISA, * MASTERCARD, * MIR, * AMEX, * UNIONPAY, * GCLOCAL. |
| p.cardholder | O | string (26) | Имя держателя карты. |
| p.isFullyAuthenticated | O | string (1) | Флаг прохождения полной аутентификации.  Возможные значения:   * Y – аутентификация завершилась со статусом Y, * N – аутентификация была неуспешной или завершилась со статусом, отличным от Y.   Параметр отсутствует, если 3DS-аутентификация не проводилась |
| p.transmissionDateTime | O | string (14) | Время и дата из авторизационного запроса в ПЦ. В большинстве случаев используется формат «MMddHHmmss». Присутствует в случае, если result\_code=1.  В платежах СБП PGA не отправляет авторизационный запрос в процессинговый центр банка, поэтому в случае сбп параметр отсутствует. |
| ts | R | string (17) | Дата и время выполнения запроса в формате «yyyyMMdd HH:mm:ss». |
| card.registered | O | string (1) | Признак регистрации карты. Параметр может принимать следующие значения:   * Y – в ходе выполнения операции была зарегистрирована карта; * N – в ходе выполнения операции регистрации карты не было.   Значение по умолчанию N. |
| card.id | C | string (32) | Идентификатор зарегистрированной банковской карты. Присутствует, если card.registered=Y. |
| card.expiry | C | string (4) | Срок действия банковской карты на момент сохранения реквизитов (YYMM). Присутствует, если card.registered=Y. |
| card.recurrent | O | string (1) | Cтатус проведения проверочной рекуррентной авторизации. Параметр может принимать следующие значения:   * Y – рекуррентный платеж по карте доступен; * N –  рекуррентный платеж по карте недоступен.   Параметр отсутствует, если не производилась проверочная рекуррентная авторизация или ее результат неопределен. |
| card.virtual | O | string (1) | Признак виртуальной карты. Параметр может принимать следующие значения:   * Y – карта виртуальная; * N –  карта обычная.   По умолчанию: N. |
| card.externalToken | O | string (255) | Внешний идентификатор карты |
| card.bankId | O | string (3) | Идентификатор банка-эмитента источника платежа |
| signature | R | string | ЭЦП банка эквайера в формате PEM (кодировано с помощью Base64). Алгоритм подписи - SHA1withRSA. Подписывается сформированный URL, начиная c “https://” (включительно) и до “&signature=” (не включая “&signature=”). |

Пример запроса регистрации платежа в магазине:

https://merchant-website.ru/RegisterPayment.php?trx\_id=0MZN2KNHVPYQ4FAV&merch\_id=IVACQ\_voyage&merchant\_trx=0MZN2KNHVPYQ4FAV&result\_code=1&amount=780000&o.order\_id=114&p.rrn=999345&p.authcode=AB2F23&p.maskedPan=545454xxxxxx5454&p.isFullyAuthenticated=Y&p.transmissionDateTime=0511170331&ts=20160511+17%3A02%3A24&signature=pIvz%2FvuTTWfA9xxUC1BAho2gHyXi6uXDgYXYLRZWX0emxL3S0xOsbGkDsLZjCD8EaDMrKa91NKWBqiY2OCGNb4CWi%2Bqjs2pRHuMzBLWEfGWcstd0g6hYMrOMqUTFpEx0uUFiqE%2FnUQzpOkHRXexACA9S7b8FdJaDkDrJKojhVdE%3D

Пример регистрации платежа в магазине – тип источника платежа Mir Pay:

https://merchant-website.ru/RegisterPayment.php?trx\_id=V0FCU4FP08GWY40V&merch\_id=GCS\_merchant2&result\_code=2&ext\_result\_code=AUTHORIZATION\_FAILED&amount=1006&account\_id=4EEA096E50948B54C32C32455AD6BCAC&o.order\_id=ord-da22087ca83a&p.rrn=121055&p.authcode=121055&p.srcType=MIR\_PAY&p.maskedPan=220220xxxxxxxxx4537&p.transmissionDateTime=1202104556&p.paymentSystem=MIR&ts=20230518 18:02:01&signature=O8cobizmEN4yYDFXVIFpi/cL5N4rC0kEt1Sygmu8mMXqYH56YKQ+DWrNaQzBFQTR4hZ1pyhiUOn8G0NtfLnmO0ifZ2GmgJ5pFwzZc/NhzaG60d7iToXAIH3nhad0pOPpeO7ekvUpwM0h7/jBWVcLw2xqw564+5zHNjGIusN36m0o3uGMzPYsQJJBKvwEw3Fdh3bZZ3NweLsem82+03tfrRWLUc1D/bf30xcRAcjKSXvEAzu8dh3OlcSpQjldWo/GbfbDkwh8b03EK0x61pXmBKf4lCvckLXgrdHCWCr+6igPkdm3IDteGvVwWO8nyxXHnIN378+O3TTop8YgjoCqbA==

Магазин формирует ответ RegisterPaymentResponse (RPRes) содержащий результат регистрации платежа (в формате XML). Параметры ответа приведены в Таблица 9.

Таблица 9 Параметры ответа RPRes

| Параметр |  | Тип | Описание |
| --- | --- | --- | --- |
| result | R | tag | Описание результата проверки. |
| result.сode | R | integer (1) | Результат регистрации платежа:  1 - результат успешный;  2 - результат неуспешный. |
| result.desc | C | string (125) | Описание результата регистрации платежа. Обязательно в случае, если result.сode=2.  Рекомендуется заполнять идентификатором транзакции, заказа и корзины в системе магазина для разрешения возможных спорных ситуаций в случае их возникновения. |

Пример ответа на запрос регистрации платежа, если магазин зарегистрировал платеж:

<register-payment-response>

<result>

<code>1</code>

<desc>OK</desc>

</result>

</register-payment-response>

* Магазин должен отправить в ответе на запрос регистрации платежа result.сode=1, если платеж не был успешным, но его результат зарегистрирован магазином.

Пример ответа на запрос регистрации платежа, если не удалось зарегистрировать платеж в магазине:

<register-payment-response>

<result>

<code>2</code>

<desc>Temporary unavailable</desc>

</result>

</register-payment-response>

Если платеж не зарегистрирован в магазине, то PGA предпримет попытки зарегистрировать результат платежа позже (количество попыток и интервал между ними конфигурируется в настройках Сервисной платформы и согласуется с магазином в процессе интеграции).

### Регистрация карты

Регистрация карт – это сохранение карты в базе данных PGA с привязкой к магазину, т.е.добавление карты в профиль магазина. Карта сохраняется в базе со статусом VERIFIED (верифицирована). Зарегистрированная карта может использоваться для платежей без повторного ввода параметров карты и для рекуррентных платежей.

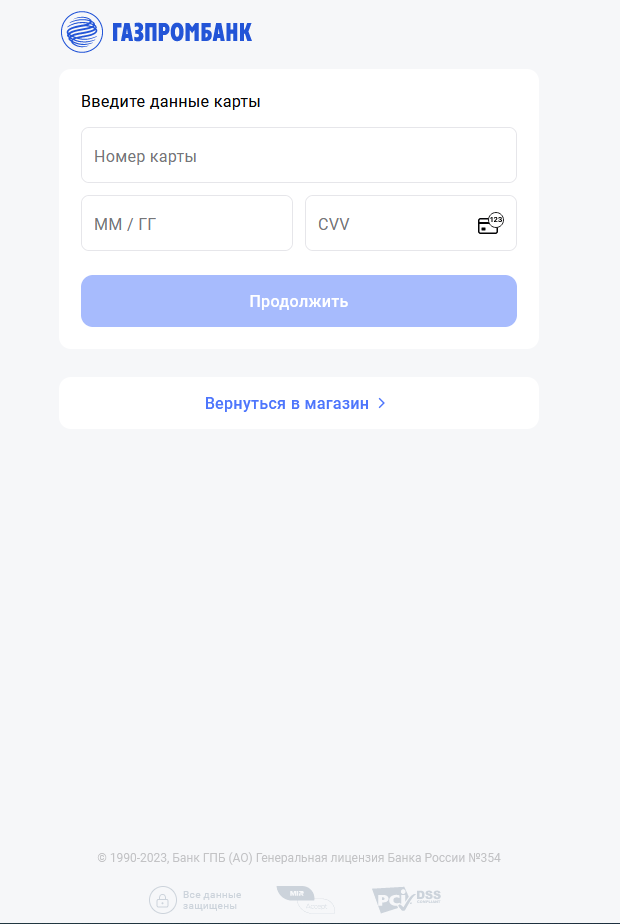


Рисунок 14 Страница регистрации карты

Пример сообщения старта регистрации карты:

3ds2.supported=true

csc=\*\*\*

expiry=1123

merchantId=GCS\_merchant2

pan=444499xxxxxx5010

params.order\_id=ord-e41114ea09c5

returnUrl=https://gazprombank.ru

state.in\_progress=no

state.redirect=post\_params

type=card

Пример сообщения результата проверки CPRes:

<?xml version="1.0" encoding="UTF-8" standalone="yes"?><payment-avail-response>

<result>

<code>1</code>

<desc>OK</desc>

</result>

<purchase>

<shortDesc>Order description</shortDesc>

<longDesc>Long order description</longDesc>

<account-amount>

<amount>1002</amount>

<fee>0</fee>

<currency>643</currency>

<exponent>2</exponent>

</account-amount>

</purchase>

<transaction-type>CardRegister</transaction-type>

</payment-avail-response>

# Приложение 1 Схема сообщений магазина CPARes, RPRes

<?xml version="1.0" encoding="UTF-8"?>  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"  
 elementFormDefault="unqualified" attributeFormDefault="unqualified">  
  
 <xs:element name="payment-avail-response">  
 <xs:complexType>  
 <xs:complexContent>  
 <xs:extension base="merchant-response">  
 <xs:all>  
 <xs:element name="result">  
 <xs:complexType>  
 <xs:all>  
 <xs:element type="xs:int" name="code"/>  
 <xs:element type="xs:string" name="desc" minOccurs="0"/>  
 </xs:all>  
 </xs:complexType>  
 </xs:element>  
 <xs:element type="xs:string" name="merchant-trx" minOccurs="0"/>  
 <xs:element type="xs:string" name="submerchant" minOccurs="0"/>  
 <xs:element name="addCardAllowed" minOccurs="0">  
 <xs:simpleType>  
 <xs:restriction base="xs:string">  
 <xs:pattern value="([YN])"/>  
 </xs:restriction>  
 </xs:simpleType>  
 </xs:element>  
 <xs:element name="transaction-type" minOccurs="0">  
 <xs:simpleType>  
 <xs:restriction base="xs:string">  
 <xs:pattern value="(CardRegister|Payment|AFT|OCT|P2P)"/>  
 </xs:restriction>  
 </xs:simpleType>  
 </xs:element>  
  
 <xs:element type="xs:string" name="RegisterRecurrent" minOccurs="0"/>  
  
 <xs:element name="card" minOccurs="0">  
 <xs:complexType>  
 <xs:all>  
 <xs:element name="id" type="xs:string" minOccurs="0"/>  
 <xs:element name="trx-id" type="xs:string" minOccurs="0"/>  
 <xs:element name="ref" type="xs:string" minOccurs="0"/>  
 <xs:element name="present" minOccurs="0">  
 <xs:simpleType>  
 <xs:restriction base="xs:string">  
 <xs:pattern value="([YN])"/>  
 </xs:restriction>  
 </xs:simpleType>  
 </xs:element>  
 </xs:all>  
 </xs:complexType>  
 </xs:element>

<xs:element name="purchase" minOccurs="0">  
 <xs:complexType>  
 <xs:all>  
 <xs:element type="xs:string" name="shortDesc" minOccurs="0"/>  
 <xs:element type="xs:string" name="longDesc"/>  
  
 <xs:element name="account-amount">  
 <xs:complexType>  
 <xs:all>  
 <xs:element type="xs:string" name="id" minOccurs="0"/>  
 <xs:element type="xs:long" name="amount"/>  
 <xs:element type="xs:long" name="fee" minOccurs="0"/>  
 <xs:element type="xs:int" name="currency"/>  
 <xs:element type="xs:int" name="exponent"/>  
 </xs:all>  
 </xs:complexType>  
 </xs:element>  
  
 <xs:element name="items" minOccurs="0">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="item" maxOccurs="unbounded">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="field" minOccurs="0"  
 maxOccurs="unbounded">  
 <xs:complexType>  
 <xs:attribute type="xs:string" name="name"/>  
 <xs:attribute type="xs:string" name="value"/>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 <xs:attribute type="xs:long" name="commission"/>  
 <xs:attribute type="xs:string" name="currency"/>  
 <xs:attribute type="xs:string" name="name"/>  
 <xs:attribute type="xs:long" name="price"/>  
 <xs:attribute type="xs:int" name="quantity"/>  
 <xs:attribute type="xs:long" name="sum"/>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 </xs:all>  
 </xs:complexType>  
 </xs:element>  
  
 <xs:element name="order-params" minOccurs="0">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="param" maxOccurs="unbounded">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element type="xs:string" name="name"/>  
 <xs:element type="xs:string" name="value"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
  
 <xs:element type="xs:string" name="primaryTrxPcid" minOccurs="0"/>  
 </xs:all>  
 </xs:extension>  
 </xs:complexContent>  
 </xs:complexType>  
 </xs:element>  
  
 <xs:element name="register-payment-response">  
 <xs:complexType>  
 <xs:complexContent>  
 <xs:extension base="merchant-response">  
 <xs:all>  
 <xs:element name="result">  
 <xs:complexType>  
 <xs:all>  
 <xs:element type="xs:int" name="code"/>  
 <xs:element type="xs:string" name="desc" minOccurs="0"/>  
 </xs:all>  
 </xs:complexType>  
 </xs:element>  
 </xs:all>  
 </xs:extension>  
 </xs:complexContent>  
 </xs:complexType>  
 </xs:element>  
  
 <xs:complexType name="merchant-response" abstract="true">  
 <xs:sequence/>  
 </xs:complexType>  
  
</xs:schema>

# Приложение 2 Расширенные коды завершения платежа

Расширенные коды завершения платежа см. в документе: [«Payguide\_Acquirer\_4\_Протокол\_OpenAPI\_Проведение\_платежей .docx».](file:///\\10.2.80.107\pwa\Shared%20Documents\_Внутренние%20проекты\Payguide%20Acquirer%202\ГПБ(ГКС)\Кастомизированные_документы_ГПБ\Payguide_Acquirer_4_Протокол_OpenAPI_Проведение_платежей_v1_35.docx)