|  |
| --- |
| **Payguide™ Acquirer**  **Протокол OpenAPI. Проведение платежей и регистрация карт** |

СОГЛАСОВАНО

| Наименование организации, предприятия | Должность исполнителя | Фамилия, имя, отчество | Подпись | Дата |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

| № версии док-та | Дата  изменения | Автор  изменений | Изменения |
| --- | --- | --- | --- |
| 1.47 | 20.10.2023 | Ерчиковская Е.В.  Симак А.Н. | * В п.3.8 добавлено описание протокола “Mir Pay”, поддерживаемого с версии 4.2.86. В Таблице 20 добавлены параметры mirPayMerchantToken и mirPayPaymentToken. Добавлена Таблица 22 Параметры запроса PUT /mirpay/accept.Обновлен пример сообщения с состоянием «offer». * В Таблица 2 «Типы запросов» добавлен запрос нового протокола “Mir Pay”. * Добавлен параметр openApiMirPaySupported в Таблица 3 Параметры запроса старта платежа п.3.3.1. * Добавлена рекомендация в п.6 проводить платёж по протоколу OpenAPI начиная с версии PGA 4.2.86. |
| 1.46 | 06.10.2023 | Ерчиковская Е.В.  Емельянов А.В. | Добавлено значение p2p. Внесены изменения:   * в п.3.3.1-Таблица 3; * в п.3.5-Таблица 15; * в п.3.12-Таблица 30; * в п.3.15-Таблица 36; * в Приложение 1. Коды ошибок выполнения запросов-Таблица 49. |
| 1.45 | 05.10.2023 | Скрежендевский В.В.  Ерчиковская Е.В. | Исключены разделы «Apple Pay», «Google Pay». |
| 1.44 | 04.10.2023 | Скрежендевский В.В. | Внесены изменения в Таблица 10, добавлены примечания к ней и уточнено описание передачи параметров магазинов-посредников |
| 1.43 | 29.09.2023 | Ерчиковская Е.В. | Добавлена колонка «Ограничения параметров» в Таблица 11 Значения параметра submerchant на старте платежа |
| Добавлен параметр params.card\_on\_file в п.3.3.1. Таблица 3 Параметры запроса старта платежа |
| 1.42 | 22.09.2023 | Ерчиковская Е.В. | Добавлен параметр raudPrevention в п.3.13,Таблица 33 Параметры результата платежа. |
| 1.41 | 23.08.2023 | Ерчиковская Е.В. | Добавлены примеры в п.3.9, Прямой режим возврата. |
| 1.40 | 15.08.2023 | Ерчиковская Е.В. | Добавлено Приложение 6.Соответствие кодов ПЦ и Open Api при завершении платежа. |
| 1.39 | 15.08.2023 | Ягур В.С. | В Таблица 53 «Статусы завершения аутентификации 3DS v2.1.0 и v2.2.0» добавлен новый статус CHALLENGE\_CANCELLED\_BY\_RBA. |
| 1.38 | 04.08.2023 | Муравицкая А.В.  Ягур В.С. | В таблицу «Сокращения и обозначения» добавлен термин «MCC». |
| В Таблица 50 «Расширенные коды завершения платежа» добавлен новый код FORMAT\_ERROR. |
| 1.37 | 10.07.2023 | Муравицкая А.В.  Ягур В.С. | Раздел 7 Протокол «Данные карт Gazprom Pay». В ответе от СПЭК добавлено поле «paymentSystem». |
| 1.36 | 23.06.2023 | Муравицкая А.В.  Ягур В.С. | В Таблица 3 «Параметры запроса старта платежа» добавлен параметр «submerchant».  Добавлена Таблица 11 «Значения параметра submerchant». |
| Добавлен термин 3DS SDK в таблицу «Сокращения и обозначения». |
| 1.35 | 14.06.2023 | Муравицкая А.В. | Обновлены примеры: Пример 1, Пример 2, Пример 3, Пример 4, Пример 5. Добавлены в примеры параметры «offerTimeout», «offerStartedAt». |
| 1.34 | 12.05.2023 | Муравицкая А.В. | В Приложение 2. Расширенные коды завершения платежа" добавлен новый код CARD\_ TEMPORARY\_BLOCKED – Карта временно заблокирована. |
| 1.33 | 25.04.2023 | Скрежендевский В.В. | Добавлен раздел 7 «Протокол «Данные карт Gazprom Pay» |
| 1.32 | 18.04.2023 | Скрежендевский В.В. | Добавлена иерархия в подраздел 3.3. Добавлены: подраздел 3.8, раздел 6. Раздел «Сокращения и обозначения» дополнен терминами: jwt и JWE. |
| 1.31 | 05.04.2023 | Муравицкая А.В. | В Таблица 33 «Параметры результата платежа» добавлен параметр «cardProduct». Пункт 3.12. |
| 1.30 | 27.03.2023 | Муравицкая А.В. | В Таблица 2 «Типы запросов» добавлен используемый при платеже запрос Mir Pay. |
| В 0 добавлена Таблица 13 «Соответствие полей авторизационного запроса и Open API». |
| Добавлено Приложение 5. Пример Open API запроса на старт In-App операции. |
| В Таблица 36 добавлен параметр «options/\*». Пункт 3.15. |
| В 0 добавлен пример Open API запроса на старт платежа Mir Pay с указанием параметров. |
| 1.29 | 23.01.2023 | Сопот И.В. | В пункт 3.3 добавлен тип платежа mir\_pay:  - в Таблица 3 добавлено значение параметра src.type: mir\_pay;  - в Таблица 4 добавлены параметры: src; src.transId |
| Добавлен пункт 0 о старте с mir\_pay |
| 1.28 | 20.12.2022 | Муравицкая А.В. | Обновлена Таблица 2 |
| 1.27 | 17.11.2022 | Муравицкая А.В. | Поддержка версионности |
| 1.26 | 22.08.2022 | Сопот И.В. | В пункт 5.3 добавлено поле externalToken |
| 1.25 | 29.07.2022 | Сопот И.В. | В пункт 5.9 добавлен параметр maskedPan |
| 1.24 | 31.05.2022 | Ерчиковская Е.В. | Дополнено описание кода ошибки UNEXPECTED\_REQUEST. Приложение 1. Коды ошибок выполнения запросов. |
| 1.23 | 24.05.2022 | Сопот И.В. | Добавлено новое поле responseCodeDetails в пункт 3.12. Для поля merchant добавлено новое подполе legalBusinessName. |
| 1.22 | 30.11.2021 | Анискин Р.А. | Обновлено описание источника платежа в пунктах 3.3 и 3.12.  Добавлен пункт 3.7. Обновлены Таблица 38, Таблица 43, Таблица 47, Таблица 50. Обновлены примеры последовательностей сообщений. |
| 1.21 | 17.09.2021 | Анискин Р.А. | Обновлены Таблица 3, Таблица 10, Таблица 38, Таблица 39, примеры последовательностей сообщений и приложения. |
| 1.20 | 03.09.2021 | Дробышевский С.В. | Удалено описание запроса сброса счётчика неуспешных попыток ввода одноразового пароля для локальных карт. |
| 1.19 | 13.07.2021 | Анискин Р.А. | Обновлено описание аутентификации по 3DS v2 в пункте 3.11 |
| 1.18 | 26.04.2021 | Анискин Р.А. | Добавлено описание типа источника платежа trx\_src |
| 1.17 | 25.01.2021 | Анискин Р.А. | Обновлено описание сообщения с состоянием iframe |
| 1.16 | 23.12.2020 | Анискин Р.А. | Добавлено описание подтверждения платежа с использованием Системы быстрых платежей. Обновлено описание источника платежа в сообщении с результатом платежа |
| 1.15 | 14.10.2020 | Анискин Р.А. | Обновлено описание запроса старта платежа и сообщения с результатом платежа |
| 1.14 | 24.09.2020 | Кушнеров Н.Р. | Добавлено описание новых расширенных кодов завершения платежа |
| 1.13 | 01.09.2020 | Кушнеров Н.Р. | Добавлено описание state: authentication. Добавлено описание запроса сброса счётчика неуспешных попыток ввода одноразового пароля для локальных карт. Обновлено описание сообщения с офертой. |
| 1.12 | 26.06.2020 | Анискин Р.А. | Добавлен код ошибки COMMISSION\_MISMATCH. Добавлен расширенный код завершения платежа CARD\_NOT\_FOUND. Обновлено описание сообщения с офертой и описание результата платежа |
| 1.11 | 19.03.2020 | Анискин Р.А. | Добавлено описание нового поля “emailReceiptEnabled” в описании магазина в пункте 3.5. Поправлен пример сообщений в пункте 6.7 |
| 1.10 | 05.03.2020 | Кушнеров Н.Р. | Добавлено описание нового поля “lang” для запроса на старт платежа и на старт верификации |
| 1.9 | 26.02.2020 | Анискин Р.А. | Добавлено описание запроса отправки уведомления с результатом платежа |
| 1.8 | 09.01.2020 | Анискин Р.А. | Добавлено описание значения “page\_js” параметра “state.redirect” |
| 1.7 | 23.12.2019 | Анискин Р.А. | Обновил версию PGA в пункте 1.1 |
| 1.6 | 12.12.2019 | Анискин Р.А. | Добавлено описание прямого режима возврата Клиента после перенаправления. |
| 1.5 | 06.12.2019 | Мамченкова М.А. | Поддержка версионности. |
| 1.4 | 30.10.2019 | Анискин Р.А. | Исправлено описание параметров запроса «Подтверждение подготовки к 3DS v2». |
| 1.3 | 29.07.2019 | Тарасенко Я.Н. | Добавлены сценарии и описание аутентификации по протоколу 3DS v2. Обновлены списки кодов ошибок. |
| 1.2 | 09.01.2019 | Анискин Р.А. | Добавлено описание регистрации карты |
| 1.1 | 14.12.2018 | Анискин Р.А. | Поддержка версионности. |
| 1.0 | 07.03.2018 | Кузеро А.М. | Начальная версия. |

Содержание

1 Общие сведения 8

1.1 Назначение документа 8

1.2 Сокращения и обозначения 8

1.3 Перечень исходных документов 9

2 Формат запросов и ответов 10

3 Платежи 11

3.1 Сценарии взаимодействия 11

3.2 Получение токена платежной транзакции 17

3.3 Старт платежа 17

3.3.1 Обычный старт платежа 17

3.3.2 Старт с помощью расшифрованного токена JWE из In-App Mir Pay 21

3.4 Обработка платежа 22

3.5 Подтверждение или отказ от платежа (оферта) 22

3.6 Подтверждение платежа с использованием Системы быстрых платежей 25

3.7 Подтверждение платежа с использованием Gazprom Pay 26

3.8 Подтверждение платежа с использованием Mir Pay 27

3.9 Перенаправления во время платежа 29

3.9.1 Обычный режим возврата 30

3.9.2 Прямой режим возврата 31

3.10 Получение текущего состояния и продолжение платежа 33

3.11 Выполнение аутентификации по протоколу 3DS v2 33

3.12 Отображение результата платежа 36

3.13 Отправка уведомления с результатом платежа 39

3.14 Скрипт state.js 40

3.15 Проведение дополнительной аутентификации 41

4 Верификация карт 43

4.1 Сценарий взаимодействия 43

4.2 Получение токена транзакции 45

4.3 Старт верификации карты 45

4.4 Обработка верификации 46

4.5 Подтверждение владения картой 47

4.6 Перенаправления 47

4.7 Получение текущего состояния и продолжение верификации карты 47

4.8 Выполнение аутентификации по протоколу 3DS v2 48

4.9 Отображение результата верификации карты 48

5 Регистрация карт 50

5.1 Сценарий взаимодействия 50

5.2 Получение токена транзакции 52

5.3 Старт регистрации карты 52

5.4 Обработка регистрации 53

5.5 Подтверждение владения картой 53

5.6 Перенаправления 53

5.7 Получение текущего состояния и продолжение регистрации карты 53

5.8 Выполнение аутентификации по протоколу 3DS v2 53

5.9 Отображение результата регистрации карты 53

6 Протокол MirPayAPI 55

7 Протокол «Данные карт Gazprom Pay» 55

8 Примеры последовательностей сообщений 56

8.1 Оплата покупки банковской картой, 3DS v1, обычный режим возврата 56

8.2 Оплата покупки банковской картой, 3DS v1, прямой режим возвратов 58

8.3 Оплата покупки банковской картой, 3DS v2.1.0, мобильное приложение 60

8.4 Оплата покупки банковской картой, 3DS v2.1.0, Web-приложение, обычный режим возврата 64

8.5 Оплата покупки банковской картой, 3DS v2.2.0, Web-приложение, прямой режим возврата, выполнение 3DS-метода 67

8.6 Выполнение рекуррентного платежа 71

8.7 Аутентификация держателя карты 72

8.8 Регистрация карты 74

Приложение 1. Коды ошибок выполнения запросов 76

Приложение 2. Расширенные коды завершения платежа 81

Приложение 3. Расширенные коды завершения верификации или регистрации карты 83

Приложение 4. Статусы завершения аутентификации 84

Приложение 5. Пример Open API запроса на старт In-App операции 86

Приложение 6. Соответствие кодов ПЦ и Open Api при завершении платежа 87

# Общие сведения

## Назначение документа

Настоящий документ описывает OpenAPI Payguide™ Acquirer – интерфейс взаимодействия внешних приложений с Сервисом для проведения платежей и регистрации карт. OpenAPI Payguide™ Acquirer является легковесным интерфейсом на базе протокола HTTPS и может быть использован в различных приложениях, включая приложения для мобильных устройств, веб-приложения, и при межсерверных взаимодействиях разного уровня сложности.

Документ описывает версию OpenApi v4 для PGA версии 4.2.86.

## Сокращения и обозначения

| № | Термин | Содержание |
| --- | --- | --- |
|  | Payguide™ Acquirer /Система | Payguide™ Acquirer – программный комплекс, обеспечивающий проведение безопасных электронных платежей с применением международных банковских карт в адрес интернет-магазинов |
|  | PAN | Primary Account Number. Номер электронной пластиковой карты |
|  | 3-D Secure | XML-протокол, который используется в качестве дополнительного защитного слоя при выполнении мобильных/электронных финансовых транзакций. Протокол разработан Visa для обеспечения безопасности интернет-платежей. Предлагается в виде услуг Verified by Visa и MasterCard SecureCode |
|  | DS | Directory Server. Внешний сервис, выполняющий маршрутизацию запросов по аутентификации платежа |
|  | Клиент | Держатель международной банковской карты, осуществляющий покупки в электронном (интернет) магазине |
|  | Эмитент | Банк-эмитент, финансовый институт, выпускающий в обращение кредитную или дебетовую карточку |
|  | Сервис | Программно-аппаратный комплекс Payguide Acquirer |
|  | КИ | Клиентский интерфейс. Это может быть мобильное приложение, WEB-страница или внешняя система, которая самостоятельно взаимодействует с клиентом |
|  | МП | Мобильное приложение |
|  | PAReq | Payer Authentication Request. Запрос на аутентификацию плательщика по протоколу 3DS v1, формируемый магазином и содержащий всю необходимую информацию о платеже |
|  | PARes | Payer Authentication Response. Ответ на запрос PAReq, получаемый магазином от Банка-эмитента, содержащий результаты аутентификации платежа |
|  | VEReq | Verify Enrollement Request. Запрос, который отправляет магазин при совершении платежа с целью выяснения участия клиента в программе 3-D Secure v1 |
|  | VERes | Verify Enrollement Response. Ответ, который получает магазин на VEReq, содержащий адрес Банка-эмитента, способного провести аутентификацию плательщика |
|  | AReq | Authentication Request Message. Запрос на аутентификацию плательщика по протоколу 3DS v2, формируемый Сервисом |
|  | ARes | Authentication Response Message. Ответ на запрос AReq |
|  | RReq | Results Request Message. Запрос от DS, содержащий результат аутентификации плательщика по протоколу 3DS v2 |
|  | RRes | Results Response Message. Ответ Сервиса на запрос RReq |
|  | СБП | Система быстрых платежей НСПК |
|  | JWT | JSON Web Token, сформированный PGA |
|  | JWE | JSON Web Encryption, подписанный и зашифрованный сервисом НСПК токен JWT2 |
|  | 3DS SDK  (3-D Secure Software Development Kit) | 3DS SDK позволяет провести 3DS-аутентификацию в мобильном приложении магазина по APP-каналу и обеспечивает:  (1) сбор необходимой для протокола MirAccept (EMV 3-D Secure) информации о мобильном устройстве Клиента (device information);  (2) поддерживает взаимодействие с компонентом Access Control Server (ACS) для аутентификации;  (3) шифрует поток данных аутентификации держателя карты «Мир».  Для аутентификации по APP-каналу версии 3DSv2, 3DS SDK должен быть сертифицирован в EMVCo и НСПК. |
|  | 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). |

## Перечень исходных документов

| ID | Название документа | Версия, Дата | Автор |
| --- | --- | --- | --- |
|  |  |  |  |

# Формат запросов и ответов

Взаимодействие осуществляется по протоколу HTTPS c использованием методов GET и POST в кодировке UTF-8.

Для всех запросов используется базовый адрес:

[https://<host>:<port>/api/<protocol\_version>**/**](https://mobile.mastercard.ru/api/v3/)**<portal\_id>/**

где  
<host> и <port> - адрес сервера,

<protocol\_version> - версия протокола. Максимальная из поддерживаемых сейчас версий - v4, минимальная - v3.

<portal\_id> - идентификатор портала, соответствующий веб-приложению, мобильному приложению или серверу сторонней системы.

Пример запроса:

POST /api/v4/01234567890ABCDEF/payment HTTP/1.1

Host: mobile.mastercard.ru

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

Content-Length: 41

param1=value1&param2=value2&param3=value3

В ответ от сервиса приложение получает JSON-структуру с необходимыми данными, например:

HTTP/1.1 200 OK

Content-Type: application/json

Content-Length: 51

Expires: Thu, 01 Dec 1994 16:00:00 GMT

Cache-Control: no-cache

{

    "param1":"value1",

    "param2":"value2"

}

Некоторые запросы являются командами сервису на выполнение различных действий и не подразумевают передачи в ответе какой-либо информации, кроме статуса выполнения команды.  
Примеры запросов-команд: запрос завершения сессии, запрос удаления карты.   
При успешном выполнении таких запросов сервис в ответе выставляет HTTP-код 200, а тело сообщения при этом отсутствует.

Например:

HTTP/1.1 200 OK

Date: Thu, 18 Feb 2016 06:42:48 GMT

Connection: keep-alive

Content-Length: 0

В случае ошибки при обработке запроса сервис выставляет один из неуспешных HTTP-кодов и возвращает информацию об ошибке (Приложение 1. Коды ошибок выполнения запросов).

Например:

HTTP/1.1 400 Bad Request

Content-Type: application/json

Content-Length: 58

Expires: Thu, 01 Dec 1994 16:00:00 GMT

Cache-Control: no-cache

{

    "error":"INVALID\_PARAMETER",

    "param":"amount"

}

В случае, если приложению не удалось получить ответ от сервиса, оно должно повторять запрос до получения ответа.

# Платежи

Сервис с помощью протокола OpenAPI обеспечивает выполнение анонимных платежей и платежей авторизованных пользователей, а также обеспечивает полную поддержку технологий 3-D Secure версий 1.0.2, 2.1.0 и 2.2.0. Сервис может применяться в решениях, не сертифицированных на соответствие стандарту безопасности PCI DSS (PA DSS). В таких случаях для ввода реквизитов банковских карт используется специальная платежная страница, размещаемая на стороне Сервиса.

## Сценарии взаимодействия

Сценарии платежа представляют собой последовательность запросов от приложения и ответов от Сервиса.

Типы запросов, используемые при платеже, приведены в Таблица 2.

Таблица Типы запросов

| Запрос | Описание | Результат |
| --- | --- | --- |
| POST /api/v4/<portal\_id>/token | Получение токена. | token |
| POST /api/v4/<portal\_id>/payment/<token>/start | Старт платежа. | state |
| POST /api/v4/<portal\_id>/payment/<token>/accept | Подтверждение платежа. | state |
| POST /api/v4/<portal\_id>/payment/<token>/mirpay/accept | Подтверждение платежа Mir Pay- протокол «Mir Pay» | state |
| POST /api/v4/<portal\_id>/payment/<token>/mirpay/prepare | Подтверждение платежа Mir Pay- протокол «Mir Pay» поддерживается с версии 4.2.86, см. пункт 3.8 | state |
| POST /api/v4/<portal\_id>/payment/<token>/decline | Отказ от платежа. | state |
| POST /api/v4/<portal\_id>/payment/<token>/3ds2-prepare/accept | Подтверждение подготовки к проведению аутентификации по протоколу 3DS 2 | state |
| POST /api/v4/<portal\_id>/payment/<token>/3ds2-prepare/reject | Отказ от проведения аутентификации по протоколу 3DS 2 | state |
| POST /api/v4/<portal\_id>/payment/<token>/authentication-confirm | Подтверждение кода аутентификации. | state |
| POST /api/v4/<portal\_id>/payment/<token>/authentication-repeat | Запрос на повторную отправку кода аутентификации | state |
| POST /api/v4/<portal\_id>/payment/<token>/authentication-decline | Отказ от прохождения аутентификации. | state |
| POST /api/v4/<portal\_id>/payment/<token> | Получение состояния платежа. | state |

Сценарий проведения платежа с использованием 3DS 1 представлен на Рисунок 1.



Рисунок 1 Сценарий платежа с использованием 3DS 1

Обозначения на диаграмме последовательности:

* Opt – опциональный блок. Присутствует при выполнении некоторого условия;
* Alt – альтернативный сценарий. Подразумевает выполнение только одного из предложенных вариантов;
* Loop – повторение действия. Подразумевает повторное выполнение действия, пока не будет выполнено некоторое условие.

Прямой поток событий сценария проведения платежа без проверки доступности в магазине (Рисунок 2):

* 1. Клиент инициирует старт платежа.
  2. UI запрашивает в PGA получение токена.
  3. UI отправляет запрос в PGA на старт платежа.
  4. Опционально. При получении ответа со статусом “in\_progress” UI отправляет запрос в PGA на получение статуса платежа.
  5. Опционально. При получении ответа со статусом “offer” UI отображает клиенту оферту.
  6. Опционально. Клиент соглашается с условиями оферты.
  7. Опционально. UI отправляет запрос в PGA на подтверждение платежа.
  8. Опционально. Клиент не соглашается с условиями оферты.
  9. Опционально. UI отправляет запрос в PGA на отклонение платежа.
  10. Опционально. При получении ответа со статусом “in\_progress” UI отправляет запрос в PGA на получение статуса платежа.
  11. Опционально. При получении ответа со статусом “redirect” UI переадресовывает клиента на страницу ACS банка-эмитента.
  12. Опционально. UI отправляет запрос в PGA на получение статуса платежа.
  13. Опционально. При получении ответа со статусом “in\_progress” UI отправляет запрос в PGA на получение статуса платежа.
  14. PGA передает в UI результат платежа.
  15. UI отображает клиенту результат платежа.

Сценарий проведения платежа с использованием 3DS 2 из МП представлен на Рисунок 2.



Рисунок 2 Сценарий платежа с использованием 3DS 2 из МП

Прямой поток событий сценария проведения платежа с использованием 3DS 2 из МП (Рисунок 3):

* 1. Клиент инициирует старт платежа.
  2. МП запрашивает в PGA получение токена.
  3. МП отправляет запрос в PGA на старт платежа.
  4. Опционально. При получении ответа со статусом “in\_progress” МП отправляет запрос в PGA на получение статуса платежа.
  5. Опционально. При получении ответа со статусом “offer” МП отображает клиенту оферту.
  6. Опционально. Клиент соглашается с условиями оферты.
  7. Опционально. МП отправляет запрос в PGA на подтверждение платежа.
  8. Опционально. Клиент не соглашается с условиями оферты.
  9. Опционально. МП отправляет запрос в PGA на отклонение платежа.
  10. Опционально. При получении ответа со статусом “in\_progress” МП отправляет запрос в PGA на получение статуса платежа.
  11. Опционально. При получении ответа со статусом “3ds2\_prepare ” МП отправляет запрос подтверждения подготовки к аутентификации с использованием 3DS2, если оно поддерживает одну из версий протокола 3DS 2, указанных Сервисом.
  12. Опционально. При получении ответа со статусом “3ds2\_prepare ” МП отправляет запрос отказа от подготовки к аутентификации с использованием 3DS 2, если оно не поддерживает ни одну из версий протокола 3DS 2, указанных Сервисом.
  13. Опционально. При получении ответа со статусом “3ds2\_challenge” МП взаимодействует с ACS банка-эмитента с целью аутентификации Клиента.
  14. Опционально. МП отправляет запрос в PGA на получение статуса платежа.
  15. Опционально. При получении ответа со статусом “in\_progress” МП отправляет запрос в PGA на получение статуса платежа.
  16. PGA передает в МП результат платежа.
  17. МП отображает клиенту результат платежа.

Сценарий проведения платежа с использованием 3DS 2 из браузера представлен на Рисунок 4.



Рисунок 3 Сценарий платежа с использованием 3DS 2 из браузера

Прямой поток событий сценария проведения платежа с использованием 3DS 2 из браузера (Рисунок 5):

* 1. Клиент инициирует старт платежа.
  2. Браузер запрашивает в PGA получение токена.
  3. Браузер отправляет запрос в PGA на старт платежа.
  4. Опционально. При получении ответа со статусом “in\_progress” Браузер отправляет запрос в PGA на получение статуса платежа.
  5. Опционально. При получении ответа со статусом “offer” Браузер отображает клиенту оферту.
  6. Опционально. Клиент соглашается с условиями оферты.
  7. Опционально. Браузер отправляет запрос в PGA на подтверждение платежа.
  8. Опционально. Клиент не соглашается с условиями оферты.
  9. Опционально. Браузер отправляет запрос в PGA на отклонение платежа.
  10. Опционально. При получении ответа со статусом “in\_progress” Браузер отправляет запрос в PGA на получение статуса платежа.
  11. Опционально. При получении ответа со статусом “3ds2\_prepare ” Браузер отправляет запрос подтверждения подготовки к аутентификации с использованием 3DS 2, если он поддерживает одну из версий протокола 3DS 2, указанных Сервисом.
  12. Опционально. При получении ответа со статусом “3ds2\_prepare ” Браузер отправляет запрос отказа от подготовки к аутентификации с использованием 3DS 2, если он не поддерживает ни одну из версий протокола 3DS 2, указанных Сервисом.
  13. Опционально. При получении ответа со статусом “3ds2\_challenge” Браузер взаимодействует с ACS банка-эмитента с целью аутентификации Клиента.
  14. Опционально. Браузер отправляет запрос в PGA на получение статуса платежа.
  15. Опционально. При получении ответа со статусом “in\_progress” Браузер отправляет запрос в PGA на получение статуса платежа.
  16. PGA передает в Браузер результат платежа.
  17. Браузер отображает клиенту результат платежа.

## Получение токена платежной транзакции

Перед началом каждого платежа приложение получает уникальный токен платежной транзакции, который используется для идентификации транзакции во всех последующих запросах, а также служит для исключения дубликатов при повторе запросов.

Запрос на получение токена:

POST /api/v4/<portal\_id>/token

В случае успешного выполнения запроса Сервис возвращает JSON-структуру, содержащую уникальный токен платежной транзакции.

Пример запроса на получение токена:

POST/api/v4/01234567890ABCDEF/token HTTP/1.1

Host: <portalname>

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

Content-Length: 0

Пример ответа:

HTTP/1.1 200 OK

Content-Type: application/json

Content-Length: 28

Expires: Thu, 01 Dec 1994 16:00:00 GMT

Cache-Control: no-cache

{

"token":"CA63E4F34DA634BC"

}

## Старт платежа

### Обычный старт платежа

Запрос на старт платежа:

POST /api/v4/<portal\_id>/payment/<token>/start

где <token> – значение токена, полученное на предыдущем шаге.

Параметры запроса приведены в Таблица 3.

Таблица Параметры запроса старта платежа

| Параметр |  | Описание |
| --- | --- | --- |
| merchantId | С | Идентификатор магазина.  Параметр является обязательным при выполнении эквайринговых платежей, потому что все такие платежи выполняются в рамках конкретного магазина. |
| accountId | O | Идентификатор счета магазина.  Необязательное поле. В некоторых случаях Сервис сам может выбрать нужный счёт среди счетов магазина. |
| paymentId | O | Идентификатор платежа.  При выполнении обычных эквайринговых платежей этот параметр не используется. Но если у магазина настроены дополнительные платёжные схемы (P2P, AFT, OCT и пр.), то параметр надо указывать для того, чтобы Сервис выбрал правильную платёжную схему. Возможные значения:   * p2p * aft * oct * gambling |
| merchantTrx | O | Идентификатор транзакции магазина. |
| description | C | Описание платежа.  Параметр является обязательным в некоторых сценариях платежа. |
| currency | С | Трехбуквенный код валюты (ISO 4217), например: RUB.  Параметр является обязательным в некоторых сценариях платежа. |
| amount | С | Сумма платежа в минорных единицах.  Параметр является обязательным в некоторых сценариях платежа. |
| commission | O | Сумма комиссии в минорных единицах. |
| src.type | C | Тип источника средств.  Возможные значения:   * card – необходимые атрибуты карты указываются в запросе; * card\_id – в запросе указывается id зарегистрированной карты; * card\_page – реквизиты банковской карты вводятся на платежной странице, размещаемой на стороне сервиса; * trx\_src – атрибуты карты нужно взять из описания источника средств ранее проведённой транзакции; * gazprom\_pay – атрибуты карты нужно взять из кошелька Gazprom Pay; * mir\_pay – атрибуты карты нужно взять из кошелька Mir Pay;   Параметр является обязательным в некоторых сценариях платежа. |
| src.\* | O | Параметры источника средств (набор и формат параметров определяется значением параметра src.type). |
| dst.type | C | Тип получателя средств. Возможные значения:   * card – необходимые атрибуты карты указываются в запросе; * card\_id – в запросе указывается id зарегистрированной карты;   Параметр является обязательным в некоторых сценариях платежа. |
| dst.\* | O | Параметры получателя средств (набор и формат параметров определяется значением параметра dst.type). |
| params.\* | O | Параметры платежа (набор и формат параметров определяется магазином). |
| params.card\_on\_file | C | Возможные значения:   * UCOF - Платеж , запущенный магазином по инициативе держателя карты (по событию); * MIT - Платеж по инициативе магазина (по расписанию).   Данное поле обязательное для заполнения, если значение поля recurrent=true |
| returnUrl | C | Адрес возврата для продолжения платежа после перенаправления.  Обязательный параметр при поддержке перенаправлений. |
| state.\* | O | Опциональные параметры, позволяющие приложению использовать специфические режимы обработки отдельных состояний платежа. |
| receiptEmail | O | Адрес, на который требуется отправить уведомление с результатом платежа. |
| preauthorize | O | Флаг выполнения предварительной авторизации, т.е. блокировки средств. Возможные значения: true/false.  По умолчанию false. |
| recurrent | O | Признак рекуррентного платежа (card not present).  Возможные значения: true/false.  По умолчанию false. |
| submerchantName | C | Название подмагазина.  Параметр обязательный для магазинов-посредников. |
| deviceFingerprint | O | Цифровой отпечаток устройства пользователя.  Например, хэш набора характеристик браузера клиента.  Передаётся в систему фрод-мониторинга. |
| 3ds2.supported | O | Признак поддержки приложением протокола 3DS 2.  Возможные значения: true/false.  По умолчанию false. |
| 3ds2.destWalletNum, 3ds2.destCardNum,  3ds2.destPhoneNum,  3ds2.destAcctNum | C | Описание получателя денежных средств при переводах на электронный кошелёк, карту, телефон или счёт.  Параметр обязательный при проведении 3DS v2 по картам МИР для некоторых категорий магазинов |
| lang | O | Язык транзакции (ISO 639-1), например, ru. |
| submerchant. \* | О | Параметры подмагазина, который проводит операции с банком через магазин-посредник (facilitator) – (см. Таблица 11). |
| openApiMirPaySupported | O | Для проведения платежа по сценарию «MirPay» можно выбрать способ взаимодействия с API. Если параметр не указан или false, для инициации MirPay используются запрос /mirpay/accept. Если равен true, то следует использовать /mirpay/prepare. Второй вариант более пердпочтительный и поддерживает возможность отменить MirPay платёж для возможности выбора другого метода оплаты. Для отмены следует использовать метод /decline. Сам /mirpay/prepare можно снова повторить после /decline. Полное описание приведено в пункте «Подтверждение платежа с использованием Mir Pay».  Возможные значения: true/false.  По умолчанию false. |

Для источника с типом card необходимо указать параметры, приведенные в Таблица 4.

Таблица Параметры источника card

|  |  |  |
| --- | --- | --- |
| Параметр |  | Описание |
| src.pan | R | Номер карты.  В случае оплаты Mir Pay это tan (Token Account Number – токенизированный номер карты). |
| src.cavv | C | Authentication Value (NSPK-InApp операции) Обязательный, когда ТСП передает в PGA параметры из расшифрованного токена JWE. |
| src.transId | C | Уникальный идентификатор NSPK-InApp операции. |
| src.expiry | R | Срок действия карты в формате ММГГ. |
| src.csc | O | Card Security Code (CVV2/CVC2). |
| src.cardholder | O | Имя держателя карты. |
| src.addToProfile | O | Флаг, обозначающий необходимость добавления карты в профиль клиента или магазина после успешного платежа.  Возможные значения: true/false. |

Сервис может принять запрос на старт платежа, в котором указаны номер карты, срок действия карты и Card Security Code (CVV2/CVC2), но для обеспечения безопасности данные параметры не передаются в сообщениях с описанием состояния платежа и сообщениях содержащих историю операций.

Для источника с типом card\_id необходимо указать параметры, приведенные в Таблица 5.

Таблица Параметры источника card\_id

| Параметр |  | Описание |
| --- | --- | --- |
| src.cardId | R | Идентификатор карты. |
| src.csc | O | Card Security Code (CVV2/CVC2). |

Для источников с типом card\_page необходимо указать параметр, приведенный в Таблица 6.

Таблица Параметр источника card\_page

| Параметр |  | Описание |
| --- | --- | --- |
| src.addToProfile | O | Флаг, указывающий, необходимость добавления карты в профиль клиента после успешного платежа. |

Для источника с типом trx\_src необходимо указать параметры, приведенные в Таблица 7.

Таблица Параметры источника trx\_src

| Параметр |  | Описание |
| --- | --- | --- |
| src.token | R | Токен платёжной транзакции, из описания источника средств которой нужно взять атрибуты карты. |
| src.csc | O | Card Security Code (CVV2/CVC2). |

Для источника с типом Gazprom\_pay дополнительные параметры указать не нужно.

Для получателя с типом card необходимо указать параметры, приведенные в Таблица 8.

Таблица Параметры получателя card

| Параметр |  | Описание |
| --- | --- | --- |
| dst.pan | R | Номер карты. |

Для получателя с типом card\_id необходимо указать параметры, приведенные в Таблица 9.

Таблица Параметры получателя card\_id

| Параметр |  | Описание |
| --- | --- | --- |
| dst.cardId | R | Идентификатор карты. |

В ответ на запрос старта платежа Сервис может вернуть сообщение об ошибке или сообщение со статусом платежа. В зависимости от значения поля state в сообщении со статусом платежа, приложение выполняет различные действия для продолжения платежа, приведенные в Таблица 10.

Таблица Значения поля state

| Значение поля “state” | Действие приложения |
| --- | --- |
| in\_progress | Через некоторое время выполнить новый запрос состояния. |
| offer | Отобразить оферту и запросить подтверждение платежа у клиента. |
| authentication | Провести аутентификацию клиента. |
| redirect | Выполнить перенаправление на указанный адрес. |
| 3ds2\_prepare | Выбрать конкретную версию протокола 3DS 2 и собрать необходимые данные о клиенте и КИ |
| 3ds2\_challenge | Начать взаимодействие с ACS для проведения явной аутентификации клиента |
| 3ds2\_info | Отобразить клиенту информационное сообщение, связанное с прохождением аутентификации |
| iframe | Отобразить на web-странице фрейм с указанными параметрами |
| result | Отобразить информацию о результате платежа. |

Для платежной схемы «Интернет-эквайринг ГПБ без CPA» при условии, что в настройках платежной схемы в Административной Консоле на уровне магазина флаг "Магазин-посредник" установлен, магазин-посредник должен передать в Open API-запросе на старт платежа следующие параметры (Таблица 11). В таком случае (значение поля submerchantName в сообщении «Старт платежа» игнорируется PGA). Если установлен флаг "Магазин-посредник" и поля с параметрами подмагазина на старте платежа отсутствуют или не заполнены, то транзакция должна завершиться ошибкой с кодом MISSING\_PARAMETER.

Таблица Значения параметра submerchant на старте платежа

| Значение параметра  “submerchant” | Действие приложения | Ограничения параметров |
| --- | --- | --- |
| submerchant.city | Город | [A-Za-z 0-9\*-\_]{1,13}\* |
| submerchant.country | Страна | [A-Z]{3} |
| submerchant.id | Идентификатор подмагазина | [A-Za-z 0-9\*-\_]{15} |
| submerchant.name | Наименование подмагазина | [A-Za-z 0-9\*-\_]{..}\*\* |
| submerchant.terminalId | Идентификатор терминала подмагазина | [A-Za-z 0-9\*-\_]{8} |
| submerchant.mcc | MCC | [0-9]{4} |
| submerchant.inn | ИНН | [0-9]{20} |

\* - Примечание. Если длина поля «submerchant.city» больше 13 символов – PGA обрежет его справа, если меньше 13 символов - заполнит справа пробелами.

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

Префикс {FacilitatorName} не боле 11 символов:

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

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

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

### Старт с помощью расшифрованного токена JWE из In-App Mir Pay

Начиная с версии PGA4.2.69 в платежную схему «Платежи без проверки доступности в магазине» добавлен функционал для реализации платежей через Сервис In-App Mir Pay. ТСП на старте передает в PGA параметры из расшифрованного токена JWE, необходимые для формирования авторизационного запроса.

На уровнях платежной схемы и платежной схемы магазина добавлены настройки:

- «Mir Pay» включен;

- «Разрешить платеж по расшифрованному токену».

В Таблица 12 указаны соответствия токена JWE и стартового запроса:

Таблица 12 Соответствия полей JWE и стартового запроса

|  |  |
| --- | --- |
| Поле JWE | Поле стартового запроса |
| tan | src.pan |
| sum | amount |
| cur | currency (с переводом числового в буквенный код) |
| tem+tey | src.expiry |
| orderId | merchantTrx |
| transId | src.transId |
| cav | src.cavv в формате HEX |

Пример 1: Запрос:

POST /api/v4/01234567890ABCDEF/payment/CA63E4F34DA634BC/start HTTP/1.0

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

merchantId=merchant123456

description=Mir Pay decrypted token without CPA

merchantTrx=123456

src.type=mir\_pay

src.pan=2203000000268822

src.expiry=3010

src.cavv=00DD025ABBD3483F41000167DB1A37230601B6

src.transId=adc3adf8-91a3-11ed-a1eb-0242ac120002

amount=1000

currency=RUB

returnUrl=https://site.ru/c2c?id=1234567890ABCDEF

В случае src.type=mir\_pay поля авторизационного запроса, заполнить согласно Таблица 13.

Таблица 13 Соответствие полей авторизационного запроса и Open API

| **Поле авторизационного запроса** | **Значение (поле Open API)** | **Описание** |
| --- | --- | --- |
| 48.42 | - | SLI, Value = 406 |
| 48.45 | src.transId | Уникальный идентификатор In-Application операции |
| 48.47 | src.cavv | Authentication Value (NSPK-InApp операции) |

Пример 2: Open API запрос на старт платежа Mir Pay:

3ds2.supported=true

merchantId=GCS\_merchant2

params.order\_id=ord-0089e1fc39d5

returnUrl=https://site.ru/c2c?id=1234567890ABCDEF

state.in\_progress=no

state.redirect=post\_params

Примечание: правила передачи параметров на старт платежа определяются в соответствии с Таблица 3 Параметры запроса старта платежа".

## Обработка платежа

При выполнении платежа в течение длительного времени, например, при обращении в банк или систему фрод-мониторинга, Сервис отправляет приложению сообщение с состоянием платежа in\_progress. Данный статус платежа означает, что платеж находится в обработке и взаимодействие с клиентом не требуется.

В Таблица 14 приведены варианты поведения Сервиса в зависимости от значения параметра state.in\_progress в запросе начала платежа.

Таблица Значения параметра state.in\_progress

|  |  |
| --- | --- |
| Значение параметра  “state.in\_progress” | Режим отображения состояния |
| no | Сервис не информирует о состоянии платежа in\_progress. |
| yes (по умолчанию) | Сервис информирует о состоянии in\_progress, если через 3 секунды после поступления запроса платеж не перешел в какое-либо состояние, кроме in\_progress. |
| js | Вместо сообщения с состоянием "in\_progress" сервис возвращает перенаправление на адрес со специальным скриптом state.js. Данный скрипт будет самостоятельно запрашивать статус платежа и выполнять редиректы.  См. раздел "Скрипт state.js". |

В случае получения от Сервиса сообщения с состоянием платежа in\_progress необходимо выполнить запрос на получение текущего состояния платежа и далее действовать в соответствии с полученным состоянием платежа.

## Подтверждение или отказ от платежа (оферта)

Сценарий платежа может предполагать необходимость отображения подробной информации о платеже для подтверждения пользователем (оферты). При получении от Сервиса состояния offer, приложение отображает страницу с подробной информацией о платеже и предлагает пользователю подтвердить платеж, либо отказаться от него.

В ответе с состоянием offer Сервис возвращает данные о платеже, приведенные в Таблица 15.

Таблица Данные платежа для оферты

| Параметр |  | Описание |
| --- | --- | --- |
| token | R | Токен. |
| paymentId | O | Идентификатор платежа, например: p2p. |
| params | O | Параметры платежа (набор и формат параметров определяется магазином). |
| currency | R | Трехбуквенный код валюты (ISO 4217). |
| amount | R | Сумма платежа или перевода в минорных единицах. |
| commission | O | Сумма комиссии в минорных единицах. |
| description | O | Описание платежа. |
| descriptionParams/\* | O | Дополнительные параметры описания платежа. Формируются сервисом в ходе выполнения платежа. |
| src/\* | O | Параметры источника средств. |
| dst/\* | O | Параметры получателя средств. |
| merchant/\* | O | Описание магазина. |
| options/\* | O | Набор опций оферты. В них передаются параметры доступных средств платежа, показа дополнительных элементов на странице оферты и т.д.  Набор опций зависит от функционала, доступного платёжной схеме, и её настроек. |
| providerAlias | R | Алиас провайдера платежа. |

Параметры описания магазина (merchant) приведены в Таблица 16.

Таблица Параметры описания магазина

| Параметр |  | Описание |
| --- | --- | --- |
| name | R | Название магазина. |
| logoUrl | R | Адрес лого магазина. |
| url | R | Адрес сайта магазина. |
| emailReceiptEnabled | O | Флаг разрешения отправки email-уведомлений с чеком. Возможные значения: true/false. Значение по умолчанию: false. |

Данные об источнике платежа (src) могут присутствовать или отсутствовать. Зависит это от бизнес-сценария платежа. Основные сценарии такие:

1. Параметры источника платежа КИ указал в запросе старта платежа. В этом случае в сообщении с состоянием offer присутствует описание источника платежа, т.е. поле src/type и остальные поля, соответствующие типу источника. Клиент, принимая оферту, соглашается не только с суммой платежа и параметрами платежа, но и с параметрами источника платежа. Изменить параметры источника платежа клиент не может.
2. Параметры источника платежа не были указаны в запросе старта платежа и сервис не смог сам их определить. В этом случае в сообщении с состоянием offer не будет описания источника платежа, т.е. не будет поля src/type. Клиент, принимая оферту, соглашается с суммой платежа и параметрами платежа, а также вводит параметры источника платежа. Поля для указания источника платежа такие же, как и в запросе старта платежа.
3. Параметры источника платежа не были указаны в запросе старта платежа, сервис смог сам их определить или получил от внешней системы. В этом случае в сообщении с состоянием offer присутствует описание источника платежа, т.е. поле src/type и остальные поля, соответствующие типу источника. Клиент, принимая оферту, соглашается не только с суммой платежа и параметрами платежа, но и с параметрами источника платежа. Изменить параметры источника платежа клиент не может. Но из-за того, что параметры источника платежа получены не от клиента, а от сервиса, в них может не хватать некоторых данных. И получить эти данные сервис может только от клиента.

Например, в случае платежа с карты PAN и срок действия карты сервис определяет сам, но значения csc у сервиса быть не может, поэтому ввести csc должен клиент на этапе оферты. КИ должен дать возможность клиенту ввеcти csc, а потом должен передать значение csc в запросе подтверждения оферты. Для этого используется параметр src.csc. Параметр src.type и остальные src-поля при этом не указываются, что является признаком ситуации "Клиент согласился с параметрами источника платежа, которые указал Сервис, он не меняет источник платежа, а только дополняет его описание".

Также в сообщении с состоянием offer может присутствовать поле src/addToProfileEnabled со значением true. Оно означает, что настройки сервиса позволяют клиенту сохранить параметры карты для последующего использования в новых платежах. Если клиент, кроме принятия оферты и ввода параметров карты, соглашается на сохранение параметров карты в сервисе, то КИ добавляет в запрос принятия оферты параметр src.addToProfile. Поле src/addToProfileEnabledпока поддерживается не всеми платёжными схемами, поэтому в некоторых случаях сохранение параметров карты может быть доступно клиенту даже тогда, когда в сообщении с состоянием offer нет этого поля.

В зависимости от значения параметра state.offer в запросе старта платежа возможны следующие режимы обработки Сервисом состояния offer, приведенные в Таблица 17.

Таблица Значения параметра state.offer

| Значение параметра  “state.offer” | Режим отображения оферты |
| --- | --- |
| never | Состояние offer не поддерживается приложением. Если по сценарию платежа требуется отображение оферты, то платеж не может быть продолжен и завершается ошибкой. |
| on\_demand (по умолчанию) | Сервис возвращает состояние offer только в тех случаях, когда необходимо дополнительное подтверждение платежа пользователем (например, после расчета комиссии). |
| always | Сервис возвращает состояние offer для подтверждения всех платежей. |

Пример 1

Пример ответа:

HTTP/1.1 200 OK

Content-Type: application/json

Content-Length: 429

Expires: Thu, 01 Dec 1994 16:00:00 GMT

Cache-Control: no-cache

{

"token": "2134234234123456",

"state":"offer",

"description”: "Test payment",

"params":{

"msisdn":"79151234567",

"order\_id": "232323"

},

“merchant”: {  
 ”name”: "Test Merchant Name",  
 “logoUrl”: "https://testmerchant.ru/logo.png",

“url”: "https://testmerchant.ru",  
 “emailReceiptEnabled”: true

},  
 "amount":"30000",

"currency":"RUB",

"providerAlias": "gcs",

"options": {

"offerTimeout":150000,

"offerStartedAt":1686312101008

}

}

В случае подтверждения платежа пользователем приложение отправляет запрос:

POST /api/v4/<portal\_id>/payment/<token>/accept

Запрос может быть без параметров, если клиент соглашается с параметрами источника платежа, указанными в сообщении с состоянием offer.

Или в запросе могут присутствовать src-поля с параметрами источника платежа, которые ввёл клиент.

Или в запросе может быть только поле src.csc, если клиент соглашается с параметрами карты-источника платежа, указанными в сообщении с состоянием offer и дополняет эти параметры значением csc карты.

Также в запросе могут присутствовать dst-поля с параметрами получателя платежа, которые ввёл клиент.

В ответ Сервис возвращает JSON-структуру с информацией для продолжения платежа.

В случае отказа пользователя от платежа приложение отправляет запрос:

POST /api/v4/<portal\_id>/payment/<token>/decline

## Подтверждение платежа с использованием Системы быстрых платежей

Если в ответе с состоянием offer Сервиса присутствуют поля options/sbpEnabled со значением true, значит для платежа доступно использование Системы быстрых платежей (СБП) (Таблица 18).

Таблица Поля секции «options»

|  |  |  |
| --- | --- | --- |
| Поле |  | Описание |
| sbpEnabled | R | Флаг доступности платежа СБП |
| sbpPayload | O | Payload зарегистрированного QR-кода |

Пример 2

Пример сообщения с состоянием «offer» с возможностью использования СБП:

|  |
| --- |
| HTTP/1.1 200 OK Content-Type: application/json  {  "token": "...",  "state": "offer",  "description": "Заказ 53820",  "params": {  "order\_id": "53820"  },  "merchant": {  "name": "Интернет-магазин Dog&Cat",  "url": "https://www.dogandcat.ru"  "logoUrl": "https://www.dogandcat.ru/logo.png",  },  "amount": 5500,  "currency": "RUB",  "providerAlias": "gcs",  "options": {  "sbpEnabled": "true",  "offerTimeout":150000,  "offerStartedAt":1686312101008  }  } |

Для подтверждения платежа в случае использования СБП используется обычный запрос подтверждения платежа с указанием типа источника платежа sbp.

Пример запроса подтверждения платежа с использованием СБП:

POST /api/v4/TESTPORTAL123/payment/1234567890ABCDEF/accept HTTP/1.1

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

src.type=sbp

После того, как клиент подтвердил платёж с использованием СБП, Сервис выполняет регистрацию нового QR-кода во внешней системе и передаёт содержимое этого QR-кода в новом сообщении с состоянием offer.

Пример 3

Пример второго сообщения с состоянием «offer» для платежа с использованием СБП:

|  |
| --- |
| HTTP/1.1 200 OK Content-Type: application/json  {  "token": "...",  "state": "offer",  "description": "Заказ 53820",  "params": {  "order\_id": "53820"  },  "merchant": {  "name": "Интернет-магазин Dog&Cat",  "url": "https://www.dogandcat.ru"  "logoUrl": "https://www.dogandcat.ru/logo.png",  },  "amount": 5500,  "currency": "RUB",  "providerAlias": "gcs",  "options": {  "sbpEnabled": "true",  “sbpPayload”: https://sbpgate.ru/123456?type=01&sum=75000”,  "offerTimeout":150000,  "offerStartedAt":1686312101008  }  } |

КИ, получив такое сообщение, должен отобразить Клиенту QR-код и предложить выполнить СБП-платёж. На этапе второй оферты Клиенту так же, как и на этапе первой оферты, доступны запросы подтверждения платежа и отказа от платежа. Оба запроса нужно выполнять без параметров.

## Подтверждение платежа с использованием Gazprom Pay

Если в ответе с состоянием offer Сервиса присутствуют поля options/gazpromPayEnabled со значением true, значит для платежа доступно использование кошелька Gazprom Pay (Таблица 19).

Таблица Поля секции «options»

| Поле |  | Описание |
| --- | --- | --- |
| gazpromPayEnabled | R | Флаг доступности кошелька Gazprom Pay |

Пример 4

Пример сообщения с состоянием «offer» с возможностью использования Gazprom Pay:

|  |
| --- |
| HTTP/1.1 200 OK Content-Type: application/json  {  "token": "...",  "state": "offer",  "description": "Заказ 53820",  "params": {  "order\_id": "53820"  },  "merchant": {  "name": "Интернет-магазин Dog&Cat",  "url": "https://www.dogandcat.ru"  "logoUrl": "https://www.dogandcat.ru/logo.png",  },  "amount": 5500,  "currency": "RUB",  "providerAlias": "gcs",  "options": {  "gazpromPayEnabled": "true",  "offerTimeout":150000,  "offerStartedAt":1686312101008  }  } |

Для подтверждения платежа в случае использования кошелька Gazprom Pay используется обычный запрос подтверждения платежа с указанием типа источника платежа gazprom\_pay.

Пример запроса подтверждения платежа с использованием СБП:

POST /api/v4/TESTPORTAL123/payment/1234567890ABCDEF/accept HTTP/1.1

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

src.type=gazprom\_pay

После того, как клиент подтвердил платёж с использованием СБП, Сервис перенаправляет его на страницу кошелька Gazprom Pay. После взаимодействия клиента с кошельком, его возврата обратно в магазин и запроса статуса платежа Сервис отвечает новым сообщением с состоянием offer. Это нужно для того, чтобы клиент подтвердил выбор карты и мог ввести csc.

Пример 5

Пример второго сообщения с состоянием «offer» для платежа с использованием кошелька Gazprom Pay:

|  |
| --- |
| HTTP/1.1 200 OK Content-Type: application/json  {  "token": "...",  "state": "offer",  "description": "Заказ 53820",  "params": {  "order\_id": "53820"  },  "merchant": {  "name": "Интернет-магазин Dog&Cat",  "url": "https://www.dogandcat.ru"  "logoUrl": "https://www.dogandcat.ru/logo.png",  },  “src”: {  “type”: “gazprom\_pay”,  “pan”: “123456\*\*\*\*\*\*1234”,  “paymentSystem”: “MASTERCARD”  },  "amount": 5500,  "currency": "RUB",  "providerAlias": "gcs",  "options": {  "gazpromPayEnabled": "true",  “srcChangeDisabled”: “true”,  "offerTimeout":150000,  "offerStartedAt":1686312101008  }  } |

На этапе второй оферты Клиенту так же, как и на этапе первой оферты, доступны запросы подтверждения платежа и отказа от платежа.

## Подтверждение платежа с использованием Mir Pay

Если в ответе с состоянием offer Сервиса присутствует поле options/mirPayLink, значит для платежа доступно использование технологии Mir Pay (Таблица 20 и Таблица 21).

Таблица Параметры запроса

| Параметр |  | Описание |
| --- | --- | --- |
| mirPayLink | O | Deep Link / Universal Link с данными In-App платежа |
| mirPayMerchantToken | О | JWT необходимый для проведения MirPay-платежа в мобильном приложении с использованием SDK от НСПК |
| mirPayPaymentToken | О | JWT необходимый для проведения MirPay-платежа в мобильном приложении с использованием SDK от НСПК |

Таблица Требования к формированию Deep Link / Universal Link

| Тип ссылки | Требования к ссылке | Допустимый формат |
| --- | --- | --- |
| Deeplink | Должна начинаться на "mirpay://".  HOST должен быть "pay.mironline.ru".  В Uri.path должно присутствовать "inapp".  Содержит данные In-Application операции (jwt). | mirpay://pay.mironline.ru/inapp/{jwt} |
| Universal Link | Должна начинаться на https://.  HOST должен быть "pay.mironline.ru".  В Uri.path должно присутствовать "inapp".  Содержит данные In-Application операции (jwt). | https://pay.mironline.ru/inapp/{jwt} |

Пример сообщения с состоянием «offer» с возможностью использования Mir Pay:

|  |
| --- |
| HTTP/1.1 200 OK Content-Type: application/json  {  "token": "...",  "state": "offer",  "description": "Заказ 53820",  "params": {  "order\_id": "53820"  },  "merchant": {  "name": "Интернет-магазин Dog&Cat",  "url": "https://www.dogandcat.ru"  "logoUrl": "https://www.dogandcat.ru/logo.png",  },  "amount": 5500,  "currency": "RUB",  "providerAlias": "gcs",  "options": {  "mirPayLink":"https://pay.mironline.ru/inapp/eyJ...qY\_Ew",  "mirPayMerchantToken":"eyJ...J4g",  "mirPayPaymentToken":"eyJ4...zDnQ"  }  } |

Набор данных для подтверждения платежа в случае использования Mir Pay отличается от обычного подтверждения платежа, поэтому для Mir Pay так же, выделен отдельный тип запроса подтверждения с отдельным адресом:

POST /api/v4/<portal\_id>/payment/<token>/mirpay/accept

PGA, получив запрос POST /api/v4/<portal\_id>/payment/<token>/mirpay/accept

от платежной страницы, понимает, что транзакция идет по потоку MirPay (In App-операция) и ждет JWE.

Начиная с версии 4.2.86 рекомендуется использовать новый протокол «Mir Pay».

Для этого, в запросе на /start платежа следует указать параметр openApiMirPaySupported=true. Это означает, что платежные страницы поддерживают новый протокол «Mir Pay»

Платёж Mir Pay инициируется запросом:

POST /api/v4/<portal\_id>/payment/<token>/mirpay/prepare

PGA отвечает статусом «offer» (mirPayInProgress=true). Транзакция переходит в ожидание JWE от НСПК.

В ответе на OpenAPI запрос со статусом «offer» дополнительно передается mirPayMerchantToken и mirPayPaymentToken, которые необходимы только для проведения MirPay-плаежа в мобильном приложении с использованием SDK от НСПК.

Если ТСП использует собственное мобильное приложение с использованием SDK от НСПК, то подтверждение платежа с криптограммой полученной от SDK следует отправить в метод:

/api/v4/<portal\_id>/payment/<token>/mirpay/accept

Пример получения данных In-Application операции JWE (запрос/ответ):

PUT /api/v4/<portal\_id>/payment/<token>/mirpay/accept

{

"orderId": <token>,

"cryptogram": <cryptogram>"  
 }

HTTP/1.1 202 Accepted

Таблица 22 Параметры запроса PUT /mirpay/accept

| Параметр |  | Описание |
| --- | --- | --- |
| orderId | R | Идентификатор заказа/транзакции ТСП |
| cryptogram | R | Криптограмма с данными In-Application платежа в формате JWE Compact Serialization |

Добавлен менеджмент сертификатов ТСП подключенных к сервису In-Application операций. Приложение «Mir Pay» отправляет в PGA запрос  
GET /api/v4/${portal}/payment/${token}/mirpay/jwks на получение конечного сертификата ТСП. PGA передает цепочку сертификатов JWKS в «Mir Pay». ТСП не использует JWKS для провведения платежа, он предназначен для приложения «Мир».

Добавлена возможность отмены ожидания криптограммы MirPay для возможности выбора другого источника платежа. Для этого следует отправить запрос:

POST /api/v4/<portal\_id>/payment/<token>/decline

Если клиент снова нажал на кнопку «Оплатить с помощью MirPay» и возникнет необходимость ожидания криптограммы, то следует повторно отправить запрос:

POST /api/v4/<portal\_id>/payment/<token>/mirpay/prepare.

## Перенаправления во время платежа

Перенаправления используются Сервисом для того, чтобы пользователь выполнил какие-либо действия на сторонней web-странице. Это может быть страница аутентификации ACS банка-эмитента, или страница ввода параметров карты. После того, как действия будут выполнены, Сервис возвращает клиента на адрес, указанный в параметре returnUrl запроса старта платежа.

В зависимости от возможностей КИ он может реализовывать различные уровни поддержки перенаправления во время платежа. Поведение Сервиса при необходимости перенаправления определяется значением параметра state.redirect в запросе старта платежа – Таблица 23.

Таблица Значение параметра state.redirect

| Значение параметра  “state.redirect” | Реализация перенаправления в приложении |
| --- | --- |
| no | Состояние redirect не поддерживается приложением. Если по сценарию платежа требуется выполнить перенаправление, то платеж не может быть продолжен и завершается ошибкой. |
| url\_only (по умолчанию) | Приложение поддерживает перенаправление по URL. Чтобы выполнить перенаправление, Сервис возвращает состояние redirect, например:  {  "state":"redirect",  "url":"https://example.com/service"  }  Приложение может выполнить перенаправление на стороне Сервиса:  HTTP/1.1 302 Found  Location: https://example.com/service  или на стороне браузера, используя JavaScript:  window.location.replace("https://example.com/service")  Мобильные приложения открывают указанный URL во встроенном браузере.  При необходимости перенаправления методом POST Сервис выполняет перенаправление на специальную страницу со скрытой формой. В момент перенаправления в браузере отображается пустая страница. |
| post\_params | Кроме перенаправления по URL, приложение самостоятельно поддерживает перенаправление методом POST.  При необходимости перенаправления методом POST Сервис передает приложению состояние redirect с URL и набором параметров формы, например:  {  "state":"redirect",  "url":"https://acs.anybank.ru/pareq", "post":{"param1":"value1","param2":"value2"}  }  Приложение создает форму с указанными параметрами и выполняет submit при помощи JavaScript. Мобильное приложение может выполнить запрос POST с указанными параметрами во встроенном браузере. |
| post\_js | Перенаправление методом POST выполняется при помощи скрипта state.js (см. [скрипт state.js](#_Скрипт_state.js)). Перенаправление по URL осуществляется так же, как для state.redirect=url\_only. |
| page\_js | Приложение поддерживает перенаправление методом POST, но не поддерживает прямые возвраты в приложение методом POST с параметрами.  При таком значении параметра Сервис будет использовать обычный режим возврата Клиента после перенаправления, даже если с Заказчиком было оговорено использование прямых возвратов. |

В ответе с состоянием redirect Сервис возвращает данные о платеже, приведенные в Таблица 24.

Таблица Данные платежа для перенаправления

| Параметр |  | Описание |
| --- | --- | --- |
| url | O | Адрес, на который нужно выполнить перенаправление |
| js | O | Адрес, по которому находится JavaScript, который нужно выполнить |
| post | O | Список параметров для POST-запроса |

Сервис поддерживает два режима возврата Клиента после перенаправления:

1. обычный - возврат Клиента на Сервис,
2. прямой - возврат Клиента в КИ.

Какой именно режим возврата использовать, определяет Заказчик PGA при написании ТЗ.

### Обычный режим возврата

При обычном режиме возврата Сервис в качестве адреса возврата Клиента указывает свой адрес. Поэтому, после того как взаимодействие Клиента и сторонней системы завершено, сторонняя система перенаправляет Клиента на адрес Сервиса. Сервис отмечает, что Клиент вернулся после перенаправления, перенаправляет Клиента на адрес returnUrl, указанный КИ в запросе старта платежа, и продолжает выполнение транзакции.

В качестве примера рассмотрим порядок перенаправлений при проведении 3DS v1 для взаимодействия PGA и магазина, использующего собственные платёжные страницы:

1. Магазин запрашивает старт платежа с токеном A123456789 через портал Portal1 и указывает в параметре returnUrl значение <http://merchant.com/return?merch_trx=T123>
2. PGA начинает платёж, доходит до выполнения 3DS и определяет, что клиента нужно перенаправить на адрес ACS <http://acs.com/3ds>
3. PGA отправляет в Магазин сообщение со статусом платежа redirect.  
   В поле url этого сообщения будет указан адрес <http://acs.com/3ds>.  
   В POST-параметре TermUrl будет указан адрес PGA <http://pga.bank.com/avi/v4/Portal1/payment/A123456789/return>
4. Магазин перенаправляет Клиента на адрес <http://acs.com/3ds>
5. Клиент проходит аутентификацию
6. По завершении аутентификации ACS перенаправляет клиента на TermUrl. Т.е. Клиент возвращается в PGA на адрес <http://service.com/avi/v4/Portal1/payment/A123456789/return>,

В параметрах Http-запроса будет указан PARes

1. PGA перенаправляет клиента на адрес Магазина <http://merchant.com/return?merch_trx=T123>. Делается это с помощью HTTP-кода 302, потому что PGA знает, что запрос прислан напрямую из браузера, а не из внешней системы Магазина, которая умеет работать по OpenApi.
2. PGA извлекает PARes из запроса, полученного от ACS на шаге 6, и продолжает транзакцию.
3. Магазин, приняв запрос на свой адрес возврата, показывает Клиенту в браузере страницу с сообщением «Платёж обрабатывается» и в фоне запрашивает у PGA статус платежа.

Пример порядка перенаправлений при проведении 3DS v2 для взаимодействия PGA и Магазина, использующего собственные платёжные страницы:

1. Магазин запрашивает старт платежа с токеном A123456789 через портал Portal1 и указывает в параметре returnUrl значение <http://merchant.com/return?merch_trx=T123>.
2. PGA начинает платёж, доходит до выполнения 3DS и определяет, что клиента нужно перенаправить на адрес ACS <http://acs.com/3ds>.
3. PGA отправляет в DS запрос AReq. В поле AReq.notificationUrl PGA указывает свой адрес для возврата. PGA получает в ответ ARes со статусом C.
4. Магазин перенаправляет Клиента на адрес ACS http://acs.com/3ds.
5. Банк-эмитент, проанализировав данные о платеже в AReq принимает решение о дальнейшей аутентификации. ACS присылает URL для перенаправления Клиента на аутентификацию.
6. DS пересылает ARes в PGA. PGA формирует CReq и отправляет в ответе "state":"iframe" URL для перенаправления Клиента на аутентификацию и Creq. В поле url этого сообщения будет указан адрес <http://acs.com/3ds>. Магазин отображает iframe в браузере.
7. Клиент проходит аутентификацию.
8. По завершении аутентификации ACS c PGA обменивается RReq /Res.
9. ACS перенаправляет клиента в PGA на свой адрес api-протокола, указанный в параметре notificationURL запроса AReq. <http://service.com/avi/v4/Portal1/payment/A123456789/return>.

В параметрах Http-запроса будет указан Cres.

1. PGA перенаправляет клиента на адрес Магазина <http://merchant.com/return?merch_trx=T123>. Делается это с помощью HTTP-кода 302, потому что PGA знает, что запрос прислан напрямую из браузера, а не из внешней системы Магазина, которая умеет работать по OpenApi.
2. PGA извлекает CRes из запроса, полученного от ACS на шаге 9, и продолжает транзакцию.
3. Магазин, приняв запрос на свой адрес возврата, показывает Клиенту в браузере страницу с сообщением «Платёж обрабатывается» и в фоне запрашивает у PGA статус платежа.

### Прямой режим возврата

При прямом режиме возврата Сервис в качестве адреса возврата Клиента указывает значение параметра returnUrl из запроса старта платежа. Поэтому, после того как взаимодействие Клиента и сторонней системы завершено, сторонняя система перенаправляет Клиента прямо в КИ. КИ отмечает, что Клиент вернулся после перенаправления, показывает ему какую-то страницу и одновременно с этим делает OpenApi-запрос в PGA, чтобы сообщить PGA, что Клиент вернулся после перенаправления, а также для того, чтобы передать в PGA HTTP-параметры, которые внешняя система указала при возврате Клиента в КИ.

В качестве примера рассмотрим порядок перенаправлений при проведении 3DS v1 для взаимодействия PGA и магазина, использующего собственные платёжные страницы:

1. Магазин запрашивает старт платежа с токеном A123456789 через портал Portal1 и указывает в параметре returnUrl значение <http://merchant.com/return?merch_trx=T123>
2. PGA начинает платёж, доходит до выполнения 3DS и определяет, что клиента нужно перенаправить на адрес ACS <http://acs.com/3ds>
3. PGA отправляет в Магазин сообщение со статусом платежа redirect.  
   В поле url этого сообщения будет указан адрес <http://acs.com/3ds>.  
   В POST-параметре TermUrl будет указан адрес Магазина <http://merchant.com/return?merch_trx=T123>
4. Магазин перенаправляет Клиента на адрес <http://acs.com/3ds>
5. Клиент проходит аутентификацию
6. По завершении аутентификации ACS перенаправляет клиента на TermUrl. Т.е. Клиент возвращается в Магазин на адрес <http://merchant.com/return?merch_trx=T123>,

В параметрах Http-запроса будет указан PARes

1. Магазин, приняв запрос на свой адрес возврата, показывает Клиенту в браузере страницу с сообщением «Платёж обрабатывается»
2. Магазин может также извлечь PARes из HTTP-запроса, проанализировать его и сохранить у себя предварительный результат аутентификации
3. Магазин в фоне отправляет в PGA OpenApi-запрос «Возврат клиента в магазин». В данном случае это будет обращение на адрес <http://pga.bank.com/avi/v4/Portal1/payment/A123456789/merchant-return>. В запрос обязательно добавляются все HTTP-параметры, полученные от ACS при возврате Клиента.
4. PGA принимает запрос от Магазина, извлекает из него PARes и продолжает выполнение транзакции.
5. Когда транзакция завершится, PGA отправит сообщение со статусом result в качестве ответа на запрос, который Магазин отправил на шаге 9. Если выполнение транзакции затянется, то PGA не станет ждать её завершения и ответит сообщением со статусом in\_progress.

Пример порядка перенаправлений при проведении 3DS v2 для взаимодействия PGA и магазина, использующего собственные платёжные страницы:

1. Магазин запрашивает старт платежа с токеном A123456789 через портал Portal1 и указывает в параметре returnUrl значение <http://merchant.com/return?merch_trx=T123>.
2. PGA начинает платёж, доходит до выполнения 3DS и определяет, что клиента нужно перенаправить на адрес ACS <http://acs.com/3ds>.
3. PGA отправляет в DS запрос Areq. В поле Areq.notificationUrl PGA указывает адрес Магазина для возврата http://merchant.com/return?merch\_trx=T123. PGA получает в ответ ARes со статусом C.
4. Магазин перенаправляет Клиента на адрес ACS http://acs.com/3ds.
5. Банк-эмитент, проанализировав данные о платеже в AReq принимает решение о дальнейшей аутентификации. ACS присылает URL для перенаправления Клиента на аутентификацию.
6. DS пересылает ARes в PGA.  
   PGA формирует CReq и отправляет в ответе "state":"iframe" URL для перенаправления Клиента на аутентификацию и Creq. В поле url этого сообщения будет указан адрес <http://acs.com/3ds>.

Магазин отображает iframe в браузере.

1. Клиент проходит аутентификацию.
2. По завершении аутентификации ACS c PGA обменивается RReq /Res.
3. ACS перенаправляет клиента на адрес, указанный в параметре notificationURL запроса AReq. Т.е. Клиент возвращается в Магазин на адрес <http://merchant.com/return?merch_trx=T123>,

В параметрах Http-запроса будет указан Cres.

1. Магазин, приняв запрос на свой адрес возврата, показывает Клиенту в браузере страницу с сообщением «Платёж обрабатывается».
2. Магазин может также извлечь CRes из HTTP-запроса, проанализировать его и сохранить у себя предварительный результат аутентификации.
3. Магазин в фоне отправляет в PGA OpenApi-запрос с CRes «Возврат клиента в магазин». В данном случае это будет обращение на адрес <http://pga.bank.com/avi/v4/Portal1/payment/A123456789/merchant-return>. В запрос обязательно добавляются все HTTP-параметры, полученные от ACS при возврате Клиента.
4. PGA принимает запрос от Магазина, извлекает из него CRes и продолжает выполнение транзакции. PGA по OpenApi получает ответ с результатом транзакции.
5. Когда транзакция завершится, PGA отправит сообщение со статусом result в качестве ответа на запрос, который Магазин отправил на шаге 12.

Недостаток прямого возврата в том, что при возврате Клиента после перенаправления КИ должен отправить в Сервис отдельный OpenApi-запрос «Возврат клиента в магазин», в котором нужно указать все параметры полученного HTTP-запроса. В случае обычного возврата КИ должен отправить в Сервис обычный запрос статуса платежа без параметров.

Достоинство прямого возврата в том, что сокращается количество переходов браузера Клиента с одного адреса на другой. При 3DS v1 будет два перехода: КИ->ACS->КИ. В случае обычного возврата при 3DS v1 будет три перехода: КИ->ACS->Сервис->КИ.

Ещё одно достоинство прямого возврата в том, что браузер Клиента не обращается напрямую на адреса Сервиса. Он обращается только на адреса КИ и адреса ACS. Некоторым заказчикам это может быть важно с точки зрения безопасности.

Формат запроса «Возврат клиента в магазин» следующий:

POST /api/v4/<portal\_id>/payment/<token>/merchant-return

В параметрах запроса нужно указать параметры, которые содержались в HTTP-запросе, с помощью которого Клиент вернулся в КИ.

В ответном сообщении Сервис возвращает JSON-структуру с информацией о текущем состоянии платежа. В зависимости от значения поля state в ответе Сервиса приложение выполняет соответствующее действие для продолжения платежа.

## Получение текущего состояния и продолжение платежа

В случае получения в ответ от Сервиса сообщения с состоянием in\_progress или возврата по адресу, указанному в параметре returnUrl после перенаправления, приложение должно продолжить выполнение платежа, выполнив запрос на получение текущего состояния платежа:

POST /api/v4/<portal\_id>/payment/<token>

В ответном сообщении Сервис возвращает JSON-структуру с информацией о текущем состоянии платежа. В зависимости от значения поля state в ответе Сервиса приложение выполняет соответствующее действие для продолжения платежа.

Веб-приложение может использовать скрипт state.js (см. [скрипт state.js](#_Скрипт_state.js)) для упрощения обработки состояния in\_progress.

## Выполнение аутентификации по протоколу 3DS v2

Если приложение указало в запросе старта платежа, что оно поддерживает 3DS v2, то Сервис может провести аутентификацию клиента по протоколу 3DS версии 2, если остальные участники платежа поддерживают эту версию. Если же хотя бы один из участников платежа не поддерживает 3DS v2, то аутентификация будет выполняться по протоколу 3DS v1.

Выполнение аутентификации по протоколу 3DS v2 начинается с перевода платежа в состояние 3ds2\_prepare. В ответе с таким состоянием Сервис возвращает данные о платеже, приведенные в Таблица 25.

Таблица Параметры сообщения с состоянием 3ds2\_prepare

| Параметр |  | Описание |
| --- | --- | --- |
| versions | R | Список версий протокола 3DS, которые поддерживаются и Сервисом, и DS, и ACS.  В этом списке может быть версия 2.1.0, версия 2.2.0 или обе версии. |
| directoryServerID | R | Идентификатор DS. Он нужен мобильному приложению для того, чтобы определить, на каких ключах шифровать данные об устройстве |

Приложение КИ, получив сообщение с состоянием 3ds2\_prepare, должно из списка версий протокола выбрать ту версию, которую оно поддерживает. А потом в запросе подтверждения подготовки к 3DS 2 указать, какую именно версию оно выбрало, а также передать Сервису описание устройства КИ. Состав информации в этом описании зависит от типа устройства и версии протокола 3DS. Названия полей описания устройства (без учёта префикса «device.») и требования к их формату определяются спецификацией 3DS EMVCo (Таблица 26).

Формат запроса подтверждения подготовки к 3DS 2:

POST /api/v4/<portal\_id>/payment/<token>/3ds2-prepare/accept

Таблица Параметры запроса подтверждения подготовки к 3DS v2

| Параметр |  | Описание |
| --- | --- | --- |
| version | R | Конкретная версия протокола 3DS v2 |
| device.channel | R | Тип канала, который выбрал КИ. Возможные значения:   * APP – канал мобильного приложения * BRW – канал Web-приложения   Для канала 3RI подготовка к 3DS не выполняется |
| device.browserAcceptHeader | R для канала BRW | Значение заголовка “Accept”. Максимальная длина 2048 |
| device.browserIP | O для канала BRW | IP-адрес клиента.  Максимальная длина 45 Значение должно соответствовать формату IPv4 или IPv6.  Если значение не указано, то будет использован remoteAddress, т.е. адрес, с которого был получен запрос. |
| device.browserJavascriptEnabled | C для канала BRW | Разрешение использования в браузере Javascript.  Значения: true\false  Параметр обязателен для версии 3DS 2.2.0.  В версии 3DS 2.1.0 этот параметр не используется. |
| device.browserJavaEnabled | C для канала BRW | Разрешение использования в браузере Java.  Значения: true\false  Параметр обязателен для версии 3DS 2.1.0 или если device.browserJavascriptEnabled имеет значение true |
| device.browserLanguage | R для канала BRW | Язык  Максимальная длина 8 |
| device.browserColorDepth | C для канала BRW | Код разрядности цвета монитора.   * Возможные значения: 1, 4, 8, 15, 16, 24, 32, 48 * Параметр обязателен для версии 3DS 2.1.0 или если device.browserJavascriptEnabled имеет значение true |
| device.browserScreenHeight | C для канала BRW | Разрешение экрана, высота.  Параметр обязателен для версии 3DS 2.1.0 или если device.browserJavascriptEnabled имеет значение true |
| device.browserScreenWidth | C для канала BRW | Разрешение экрана, ширина.  Параметр обязателен для версии 3DS 2.1.0 или если device.browserJavascriptEnabled имеет значение true |
| device.browserTZ | C для канала BRW | Разница во времени по сравнению с UTC, в минутах  Параметр обязателен для версии 3DS 2.1.0 или если device.browserJavascriptEnabled имеет значение true |
| device.browserUserAgent | R для канала BRW | Значение заголовка “User-Agent”. Максимальная длина 2048 |
| challengeWindowSize | R для канала BRW | Код размера окна при взаимодействии с клиентом в рамках challenge.  Возможные значения: 01, 02, 03, 04, 05 |
| iframeReturnUrl | O | Адрес для возврата клиента из PGA в магазин после выполнения iframe-редиректа на ACS. Если параметр не указан, то используется значение параметра returnUrl из запроса старта платежа |
| device.sdkAppID | R для канала APP | Идентификатор инсталляции/сборки SDK.  Длина 36 |
| device.sdkEncData | R для канала APP | Зашифрованные данные с описанием устройства. Максимальная длина 64000 |
| device.sdkEphemPubKey | R для канала APP | Josn-объект с параметрами публичного компонента ключа для установки безопасного соединения SDK-ACS.  Максимальная длина 256 |
| device.sdkMaxTimeout | R для канала APP | Максимальное для SDK время на выполнения challenge в минутах. Диапазон значений: 5-30 |
| device.sdkReferenceNumber | R для канала APP | Идентификатор версии SDK и вендора SDK.  Длина 32 |
| device.sdkTransID | R для канала APP | Идентификатор транзакции SDK.  Длина 36 |
| device.sdkInterface | R для канала APP | Тип интерфейса, который поддерживает SDK для выполнения challenge.  Возможные значения: 01, 02, 03 |
| device.sdkUiType | R для канала APP | Список типов элементов интерфейса пользователя, которые поддерживает SDK. Возможные значения элементов списка:  01, 02, 03, 04 - для интерфейса 01;  01, 02, 03, 04, 05 - для других интерфейсов |

Если приложение не поддерживает ни одну из версий протокола 3DS v2, указанных Сервисом, то оно должно отправить запрос на отказ от подготовки к аутентификации:

POST /api/v4/<portal\_id>/payment/<token>/3ds2-prepare/reject

После того, как подготовка к аутентификации по протоколу 3DS v2 завершена, Сервис выполняет саму аутентификацию. Если аутентификация проходит по Frictionless-схеме (скрытая аутентификация), то для её выполнения никаких дополнительных взаимодействий с клиентом не потребуется. Если же аутентификация проходит по Challenge-схеме (явная аутентификация), то Сервису потребуется выполнить дополнительно одно-два взаимодействия с приложением КИ.

Выполнение явной аутентификации клиента проходит по-разному для мобильного приложения (канал APP) и Web-приложения (канал BRW).

В случае МП Сервис на запрос подготовки к 3DS v2 отвечает сообщением с состоянием 3ds2\_challenge. Данные из этого сообщения нужно передать в SDK для начала взаимодействия SDK и ACS (Таблица 27).

Таблица Параметры сообщения с состоянием 3ds2\_challenge

| Параметр |  | Описание |
| --- | --- | --- |
| 3dsServerTransID | R | Идентификатор транзакции 3DS Server |
| acsTransID | R | Идентификатор транзакции ACS |
| acsReferenceNumber | R | Идентификатор версии ACS и вендора ACS. |
| acsSignedContent | R | Адрес ACS и ключи для установки безопасного соединения SDK-ACS |

После того, как МП закончит выполнение явной аутентификации, оно должно запросить у Сервиса статус платежа. Если в ответ будет получено сообщение со статусом 3ds2\_challenge, значит Сервис ещё не получил от DS сообщение с результатом аутентификации и считает, что аутентификация ещё не завершена. Если в ответ будет получено сообщение со статусом in\_progress, значит, аутентификация уже завершена и Сервис перешёл к следующему шагу в сценарии платежа. А если в ответ будет получено сообщение со статусом result, значит, завершена не только аутентификация, но и весь платёж.

В случае Web-приложения Сервис на запрос подготовки к 3DS v2 отвечает сообщением с состоянием iframe. Данные из этого сообщения нужно использовать для встраивания в Web-страницу фрейма, с помощью которого будет выполняться взаимодействие клиента и ACS (Таблица 28).

Таблица Параметры сообщения с состоянием iframe

| Параметр |  | Описание |
| --- | --- | --- |
| iframeMode | R | Режим показа фрейма. Возможные значения: HIDDEN – скрытый, SIZED – фрейм с указанием размеров, FULL\_SCREEN - полноэкранный фрейм |
| iframeWidth | C | Ширина фрейма. Параметр указывается для SIZED-фрейма |
| iframeHeight | C | Длина фрейма. Параметр указывается для SIZED-фрейма |
| url | R | Адрес, с которого нужно получить данные |
| post | R | Список параметров POST-запроса. При выполнении явной аутентификации в списке будет параметр creq.  При выполнении 3DS-метода в списке будет параметр threeDSMethodData. |

Завершение явной аутентификации Web-приложение может отследить по факту обращения на адрес iframeReturnUrl, указанный на этапе подготовки к 3DS v2. После того, как аутентификация завершена, приложение должно запросить у Сервиса статус платежа.

В ходе аутентификации, проводимой для Web-приложения, ACS может дополнительно собирать данные с браузера клиента. Для этого браузер должен встроить на страницу скрытый фрейм, подгрузив данные с определённого адреса на стороне ACS. В спецификации 3DS этот процесс сбора данных называется “3DS-метод”. На уровне протокола OpenApi он реализуется с помощью состояния платежа iframe, описанного выше.

Когда выполнения 3DS-метода будет завершено, Web-приложение должно заново запросить статус платежа. Отследить завершение выполнения 3DS-метода приложение может по обращению на адрес iframeReturnUrl.

У состояния платежа iframe много общего с состоянием redirect. Состояние redirect требует перенаправления браузера Клиента на адрес внешней системы, а состояние iframe требует отображения в браузере контента, полученного от внешней системы. В обоих случаях происходит взаимодействие браузера с внешней системой. Поэтому можно сказать, что состояние iframe – это частный случай состояния redirect, т.е. это iframe-перенаправление.

При iframe-перенаправлении тоже есть два режима возврата Клиента:

1. обычный.
2. прямой.

При обычном режиме возврата Клиент из внешней системы возвращается в Сервис, а из Сервиса перенаправляется в КИ.

При прямом режиме возврата Клиент из внешней системы возвращается в КИ, а КИ после этого отправляет в Сервис запрос «Возврат клиента в магазин» для передачи HTTP-параметров возврата. Формат запроса описан в пункте 3.9.2.

Порядок сообщений при выполнении платежа с аутентификацией 3DS v2 в прямом режиме возврата и с выполнением 3DS-метода приведён в примере 8.5.

В версии 3DS 2.2.0 к Frictionless-схеме и Challenge-схеме аутентификации добавилась Decoupled-схема (раздельная аутентификация). Эта схема предполагает, что ACS после получения AReq отвечает сообщением Ares со статусом «D», потом самостоятельно (без участия 3DS Server) проводит аутентификацию клиента и отправляет запрос RReq с результатом этой аутентификации.

Для того, чтобы клиент понял, какие действия ему нужно выполнить для прохождения аутентификации, ACS указывает в Ares текст, который нужно показать клиенту. Этот текст Сервис передаёт в КИ с помощью сообщения с состоянием 3ds2\_info.

Таблица Параметры сообщения с состоянием 3ds2\_info

| Параметр |  | Описание |
| --- | --- | --- |
| 3dsServerTransID | R | Идентификатор транзакции 3DS Server |
| acsTransID | R | Идентификатор транзакции ACS |
| cardholderInfo | R | Текстовое сообщение, которое нужно показать клиенту |

КИ, получив сообщение с состоянием 3ds2\_info. должен показать клиенту текстовое сообщение, а потом через некоторое время запросить статус платежа.

Сервис в ответ на запрос статуса платежа возвращает сообщение с состоянием 3ds2\_info до тех пор, пока не получит от DS запрос RReq.

## Отображение результата платежа

При получении от Сервиса состояния result приложение отображает страницу с результатом платежа. В ответе Сервиса содержатся параметры, приведенные в Таблица 30.

Статусы завершения аутентификации показаны в Приложение 4. Статусы завершения аутентификации.

Таблица Описание результата платежа

| Параметр |  | Описание |
| --- | --- | --- |
| token | R | Токен, уникальный идентификатор платежа. |
| startedAt | R | Время начала операции (Unix time в ms) |
| finishedAt | R | Время завершения операции (Unix time в ms) |
| paymentId | O | Идентификатор платежа, например: p2p. |
| params | O | Параметры платежа (набор и формат параметров определяется магазином). |
| currency | C | Трехбуквенный код валюты (ISO 4217). Параметр может отсутствовать, если платёж был неуспешным. |
| amount | C | Сумма платежа в минорных единицах. Параметр может отсутствовать, если платёж был неуспешным. |
| commission | O | Сумма комиссии в минорных единицах. |
| description | O | Описание платежа. |
| descriptionParams/\* | O | Дополнительные параметры описания платежа. Формируются Сервисом в ходе выполнения платежа. |
| src/\* | O | Параметры источника средств. |
| dst/\* | O | Параметры получателя средств. |
| merchant/\* | O | Описание магазина. В нем может быть подполе legalBusinessName. – юридическое наименование организации. |
| refundable | O | Возможно ли выполнение возврата (или отмены платежа) по данной транзакции (true/false) по инициативе клиента. По умолчанию false.  Все эквайринговые платежи являются анонимными, т.е. их выполняют неаутентифицированные клиенты. Для анонимных платежей возврат средств и отмена платежа по инициативе клиента недоступны. |
| portalType | R | Тип портала, из которого был инициирован платёж.  Возможные значения:   * MOBILE * SMS * USSD * WEB * PROVIDER * AUTO\_PAYMENT |
| type | R | Тип платежа.  Возможные значения:   * PAYMEN - оплата товара/услуги * CARD\_TO\_CARD - перевод с карты на карту * CARD\_TO\_CASH - перевод с карты в наличные * ACQUIRING - интернет-эквайринг * CARD\_TO\_ACCOUNT - перевод с карты на счёт * ACCOUNT\_TO\_CARD - перевод со счёта на карту * GAMBLING – пополнение счёта магазина и ввод со счёта магазина на карту * TEST\_PAYMENT – тестовый платёж |
| result/\* | R | Детали результата платежа |
| options/\* | O | Набор опций результата платежа.  В них передаются параметры показа дополнительных элементов на странице результата платежа.  Набор опций зависит от функционала, доступного платёжной схеме, и её настроек. |

Параметры источника средств (src) приведены в Таблица 31.

Таблица Параметры источника средств

| Параметр |  | Описание |
| --- | --- | --- |
| type | R | Тип платежного инструмента. Возможные значения:   * card – банковская карта * sbp – платёж с помощью Системы быстрых платежей * gazprom\_pay – платёж с помощью кошелька Gazprom Pay |
| pan | C | Маскированный PAN карты. Используется для платежных инструментов card, gazprom\_pay. |
| cardId | С | Идентификатор карты. Используется для платежных инструментов card. Присутствует, если платёж проводился с помощью сохранённой (зарегистрированной) карты или сохранение карты было выполнено в ходе платежа |
| title | С | Имя карты. |
| addedToProfile | О | Признак того, что карта была добавлена в профиль магазина (зарегистрирована) в ходе выполнения платежа (только для платежных инструментов card). Возможные значения: true/false. По умолчанию параметр имеет значение false. |
| recurrentSupported | C | Флаг поддержки картой рекуррентных платежей. Используется только при выставленном флаге добавления карты в профиль.  Возможные значения: true/false. Если поле отсутствует, значит, проверка на рекурренты не проводилась или она закончилась неопределённым результатом. |
| paymentSystem | C | Идентификатор платёжной системы карты. Возможные значения:   * VISA, * MASTERCARD, * MIR, * AMEX, * UNIONPAY, * GCLOCAL |
| account | C | Идентификатор клиента в Системе быстрых платежей. Только для платежного инструмента sbp. |

Параметры получателя средств (dst) приведены в Таблица 32.

Таблица Параметры получателя средств

| Параметр |  | Описание |
| --- | --- | --- |
| type | R | Тип платежного инструмента. Возможные значения:   * card – банковская карта |
| pan | C | Маскированный PAN карты. Только для платежного инструмента card. |
| cardId | С | Идентификатор карты. Только для платежного инструмента card. Присутствует, если платёж проводился на сохранённую (зарегистрированную) карту |
| title | С | Имя карты. |

Параметры результата платежа (result) приведены в Таблица 33.

Таблица Параметры результата платежа

| Параметр |  | Описание |
| --- | --- | --- |
| status | R | Статус платежной транзакции:   * SUCCESS - транзакция завершена успешно; * REFUND - транзакция завершена успешно, а далее был выполнен возврат средств или отмена; * INTERIM\_SUCCESS - первая фаза транзакции завершена успешно; * DECLINED - клиент отказался от продолжения транзакции; * FAILED - транзакция завершена неуспешно; * UNKNOWN - статус транзакции не определён. |
| extendedCode | O | Расширенный код завершения платежной транзакции (см. Приложение 2. Расширенные коды завершения платежа). |
| responseCode | O | Код процессинга. |
| responseCodeDetails | O | Детализация кода ответа процессинга. |
| isFullyAuthenticated | O | Флаг проведения успешного полного 3DS.  Значение true означает, что результат был VERes=Y / PARes=Y. Используется в некоторых сценариях платежа. Поле отсутствует, если 3DS-аутентификация не проводилась. |
| authorizationTime | O | Дата и время выполнения авторизационного запроса в формате "MMddHHmmss".  Используется в некоторых сценариях платежа. Поле отсутствует, если запрос на авторизацию не выполнялся. |
| trxId | O | Идентификатор платежной транзакции для показа клиенту. По соображениям безопасности показывать клиенту полный токен транзакции нельзя. Поэтому специально для показа клиенту Сервис формирует укороченный идентификатор транзакции, который состоит из десяти первых символов токена. |
| rrn | O | Код RRN. |
| approvalCode | O | Код авторизации. |
| orderStatus | O | Статус заказа.  Возможные значения:   * UNKNOWN - статус заказа неизвестен (или нет дополнительного статуса у данного заказа); * REGISTERED - заказ зарегистрирован у продавца (в магазине); * RECEIVED - заказ выдан получателю; * REGISTRATION\_ATTEMPTS\_EXHAUSTED - исчерпаны попытки регистрации заказа у продавца (в магазине); * PROCESSING\_REFUND - заказ находится на стадии возврата. В случае успеха примет значение RECEIVED. |
| orderStatusChangedAt | O | Дата и время последнего изменения статуса заказа (Unix time в ms). |
| limit | O | Описание лимита. Поле присутствует, если операция была отклонена из-за превышения лимитов. |
| limit/type | R | Тип лимита:   * COUNT\_BY\_CUSTOMER - лимит на количество операций для клиента; * SUM\_BY\_CUSTOMER - лимит на общую сумму операций для клиента; * COUNT\_BY\_SRC\_PAN - лимит на количество операций для карты-источника. |
| limit/period | R | Период, который охватывается лимитом:   * DAY - дневной лимит; * WEEK- недельный лимит; * MONTH - месячный лимит. |
| limit/maxAmount | R | Максимальная сумма по установленному лимиту. |
| limit/remainingAmount | R | Сумма, оставшаяся до достижения установленного лимита. |
| limit/timeToRenew | O | Время до обновления состояния лимитов, в секундах. |
| cardProduct | O | Код карточного продукта, полученный от НСПК.  Значение кода карточного продукта определяется Стандартом ПС МИР. |
| fraudPrevention | O | Значение тега, раскрывающего информацию о причине неуспешного результата платежа |

Пример ответа:

HTTP/1.1 200 OK

Content-Type: application/json

Content-Length: 592

Expires: Thu, 01 Dec 1994 16:00:00 GMT

Cache-Control: no-cache

{

"token": "FDBBDE63BA123456",

"state":"result",

"description": "Test payment",  
 "params":{

"msisdn":"79151234567",

"order\_id": "121212"

},

"merchant": {  
 "name": "Test Merchant Name ",

"legalBusinessName": "Legal Business Name",  
 "logoUrl": "https://testmerchant31.ru/logo.png",

"url": "https://testmerchant31.ru",

“emailReceiptEnabled”: true  
 },  
 "src":{

"type":"card",

"pan":"531067xxxxxx0210",

“paymentSystem”: “MASTERCARD”

},

"amount":"30000",

"currency":"RUB",

"result":{

"status":"SUCCESS",

"extendedCode":"OK",

"responseCode":"000",

"responseCodeDetails": "Successful",

"trxId":"FDBBDE63BA",

"rrn":"436702609598",

"approvalCode":"024923"

"isFullyAuthenticated": "true",

"authorizationTime": "0110122234"

}

"portalType": "WEB",  
 "type": "ACQUIRING"

}

## Отправка уведомления с результатом платежа

В запросе старта платежа можно указать email-адрес, на который требуется отправить уведомление с результатом платежа. Если этот адрес указан и если отправка уведомлений с результатом платежа разрешена настройками Сервиса, то Сервис после завершения платежа автоматически сформирует сообщение с его результатом и отправит его на указанный email-адрес.

Кроме того, отправить уведомление после завершения платежа можно по отдельному запросу (Таблица 34). Формат у этого запроса такой:

POST /api/v4/<portal\_id>/payment/<token>/send-receipt

Таблица Параметры запроса отправки уведомления

| Параметр |  | Описание |
| --- | --- | --- |
| email | O | Email-адрес, на который нужно отправить электронное письмо, содержащее результат платежа |
| msisdn | O | Номер телефона, на который нужно отправить SMS с результатом платежа |

Если отправка уведомлений разрешена, и транзакция имеет подходящий статус, то Сервис выполнит отправку уведомления на указанный email-адрес и/или номер телефона.

В ответ на запрос Сервис возвращает JSON-структуру, содержащую информацию о результате отправки уведомлений (Таблица 35).

Таблица Поля ответа на запрос отправки уведомления

| Параметр |  | Описание |
| --- | --- | --- |
| emailSent | O | Результат отправки уведомления на email-адрес. true – отправка выполнена успешно,  false – отправка выполнена неуспешно. Отсутствие поля означает, что отправка уведомления на email-адрес не выполнялась. |
| msisdnSent | O | Результат отправки SMS-уведомления на номер телефона. true – отправка выполнена успешно,  false – отправка выполнена неуспешно. Отсутствие поля означает, что отправка SMS-уведомления на номер телефона не выполнялась. |

Пример запроса на отправку уведомления:

POST/api/v4/01234567890ABCDEF/payment/234234234234/send-receipt HTTP/1.1

Host: <hostname>

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

Content-Length: 18

email=test@test.com

Пример ответа:

HTTP/1.1 200 OK

Content-Type: application/json

Content-Length: 28

Expires: Thu, 01 Dec 1994 16:00:00 GMT

Cache-Control: no-cache

{

"emailSent":true

}

## Скрипт state.js

Сервис предоставляет специальный скрипт state.js, позволяющий упростить обработку в приложении некоторых состояний платежа. Для использования скрипта в запросе начала платежа необходимо указать state.in\_progress=js и state.redirect=post\_js. Параметр returnUrl при этом является обязательным:

POST /api/v4/01234567890ABCDEF/payment/CA63E4F34DA634BC/start HTTP/1.1

Host: <portalname>

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

Content-Length: 198

params.msisdn=79151234567&amount=20000&currency=RUB&src.type=card\_pa ge&state.redirect=post\_js&state.in\_progress=js&returnUrl=https%3A%2F%2Fsite.ru%2Ft opup%3Fid%3DCA63E4F34DA634BC

Для указанных состояний платежа Сервис возвращает параметр js, который содержит полный URL скрипта state.js для текущего платежа:

HTTP/1.1 200 OK

Content-Type: application/json

Content-Length: 42

Expires: Thu, 01 Dec 1994 16:00:00 GMT

Cache-Control: no-cache

{

"state":"in\_progress",

"js":"https:/WEB\_PORTAL/web/payment/CA63E4F34DA634BC/state.js"

}

При обработке полученного состояния приложение отображает страницу «Платеж обрабатывается…», которая включает скрипт, загружаемый с указанного URL:

<script src=”https:/WEB\_PORTAL/web/payment/CA63E4F34DA634BC/state.js”>

</script>

Дальнейшая обработка, включая опрос текущего состояния платежа и все необходимые перенаправления, выполняется скриптом.

## Проведение дополнительной аутентификации

Сценарий платежа может предполагать необходимость проведения дополнительной аутентификации клиента. Это может быть, например, отправка клиенту кода аутентификации по безопасному каналу связи и предложение ввести этот код на платёжной странице.

При получении от Сервиса состояния «authentication» приложение должно отобразить страницу для подтверждения кода аутентификации, опционально с подробной информацией о платеже и предложить пользователю ввести код, запросить его повторную отправку или отказаться от подтверждения (Таблица 36).

Таблица Данные платежа для аутентификации

| Параметр |  | Описание |
| --- | --- | --- |
| token | R | Токен. |
| paymentId | O | Идентификатор платежа, например: p2p. |
| params/\* | O | Параметры платежа (набор и формат параметров определяется магазином). |
| currency | R | Трехбуквенный код валюты (ISO 4217). |
| amount | R | Сумма платежа или перевода в минорных единицах. |
| commission | R | Сумма комиссии в минорных единицах. |
| Description | O | Описание платежа. |
| descriptionParams/\* | O | Дополнительные параметры описания платежа. Формируются сервисом в ходе выполнения платежа. |
| src/\* | R | Параметры источника средств. |
| dst/\* | O | Параметры получателя средств. |
| merchant/\* | O | Описание магазина. |
| merchantId | O | Идентификатор магазина, в адрес которого направлен платеж. |
| providerAlias | R | Алиас провайдера платежа |
| options/\* | O | Набор опций. В них передаются параметры доступных средств платежа, показа дополнительных элементов на странице аутентификации и т.д.  Набор опций зависит от функционала, доступного платёжной схеме, и её настроек. |

Пример ответа

HTTP/1.1 200 OK

Date: Tue, 25 Jul 2017 14:20:34 GMT

Connection: keep-alive

Content-Length: 413

Content-Type: application/json; charset=UTF-8

{

"token":"NH6A64CMIEO9R3EJ",

"state":"authentication",

"paymentId”:”p2p”,

"params":{

"order\_id":"ord-12345678"

},

"src":{

"type":"card",

"pan":"440564xxxxxx6374",

},

"dst":{

"type":"card",

"pan":"545454xxxxxx5454",

},

"amount":50000,

"commission":500,

"currency":"GEL",

"providerAlias":"bog"

}

Пример запроса на подтверждение кода

POST /api/v4/<portal\_id>/payment/<token>/authentication-confirm HTTP/1.1

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

Host: localhost:8080

Connection: keep-alive

Content-Length: 13

authCode=0182

Пример запроса на повторную отправку кода

POST /api/v4/<portal\_id>/payment/<token>/authentication-repeat HTTP/1.1

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

Host: localhost:8080

Connection: keep-alive

Content-Length: 0

Пример запроса на отказ от аутентификации

POST /api/v4/<portal\_id>/payment/<token>/authentication-decline HTTP/1.1

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

Host: localhost:8080

Connection: keep-alive

Content-Length: 0

# Верификация карт

Верификация карты – это проверка того, что клиент действительно является владельцем указанной карты. Возможны разные варианты проверки:

* проведение 3DS-аутентификации;
* резервирование случайной суммы;
* резервирование двух случайных сумм.

По результатам верификации карта не сохраняется в профиле магазина.

Если в ходе верификации была выполнена проверочная авторизация, то после верификации Сервис выполняет отмену авторизации.

## Сценарий взаимодействия

Сценарий верификации карты с точки зрения последовательности запросов во многом совпадает со сценарием платежа.

Типы запросов, используемые при верификации, приведены в Таблица 37.

Таблица Типы запросов

| Запрос | Описание | Результат |
| --- | --- | --- |
| POST /api/v4/<portal\_id>/token | Получение токена. | token |
| POST /api/v4/<portal\_id>/card/verification/<token>/start | Старт верификации. | state |
| POST /api/v4/<portal\_id>/card/verification/<token>/confirm | Подтверждение верификации | state |
| POST /api/v4/<portal\_id>/card/verification/<token>/cancel | Отказ от завершения верификации | state |
| POST /api/v4/<portal\_id>/ card/verification /<token>/3ds2-prepare/accept | Подтверждение подготовки к проведению аутентификации по протоколу 3DS 2 | state |
| POST /api/v4/<portal\_id>/ card/verification /<token>/3ds2-prepare/reject | Отказ от проведения аутентификации по протоколу 3DS 2 | state |
| POST /api/v4/<portal\_id>/card/verification/<token> | Получение состояния операции | state |

Сценарий верификации карты представлен на Рисунок 4.



Рисунок 4 Сценарий верификации карты

Обозначения на диаграмме последовательности:

* Opt – опциональный блок. Присутствует при выполнении некоторого условия;
* Alt – альтернативный сценарий. Подразумевает выполнение только одного из предложенных вариантов;
* Loop – повторение действия. Подразумевает повторное выполнение действия, пока не будет выполнено некоторое условие.

Прямой поток событий сценария проведения верификации без проверки доступности в магазине:

1. Клиент инициирует старт верификации карты.
   1. UI запрашивает в PGA получение токена.
   2. UI отправляет запрос в PGA на старт верификации карты.
   3. Опционально. При получении ответа со статусом “in\_progress” UI отправляет запрос в PGA на получение статуса верификации.
   4. Опционально. При получении ответа со статусом “confirm” UI отображает клиенту страницу для ввода проверочных данных.
   5. Опционально. Клиент вводит проверочные данные.
   6. Опционально. UI отправляет запрос в PGA на подтверждение владения картой.
   7. Опционально. Клиент отказывается от ввода проверочных данных.
   8. Опционально. UI отправляет запрос в PGA на прекращение верификации карты.
   9. Опционально. При получении ответа со статусом “in\_progress” UI отправляет запрос в PGA на получение статуса верификации.
   10. Опционально. При получении ответа со статусом “redirect” UI переадресовывает клиента на страницу ACS банка-эмитента.
   11. Опционально. UI отправляет запрос в PGA на получение статуса верификации.
   12. Опционально. При получении ответа со статусом “in\_progress” UI отправляет запрос в PGA на получение статуса верификации.
   13. PGA передает в UI результат верификации карты.
   14. UI отображает клиенту результат верификации карты.

## Получение токена транзакции

Перед началом каждой верификации приложение получает уникальный токен транзакции верификации карты, который используется для идентификации транзакции во всех последующих запросах, а также служит для исключения дубликатов при повторе запросов.

Формат запроса и ответа полностью совпадает с форматом, описанным в пункте 3.2.

## Старт верификации карты

Запрос на старт верификации карты:

POST /api/v4/<portal\_id>/card/verification/<token>/start

где <token> – значение токена, полученное на предыдущем шаге.

Параметры запроса приведены в Таблица 38.

Таблица Параметры запроса старта верификации

| Параметр |  | Описание |
| --- | --- | --- |
| merchantId | R | Идентификатор магазина. |
| accountId | С | Идентификатор счета магазина. Необязательное поле. В некоторых случаях Сервис сам может выбрать нужный счёт среди счетов магазина |
| operationId | O | Идентификатор операции. В большинстве случаев этот параметр не используется. Но если у магазина настроены дополнительные схемы регистрации или верификации карт, то параметр надо указывать для того, чтобы Сервис выбрал правильную схему. Возможные значения:   * virtual |
| merchantTrx | O | Идентификатор транзакции магазина. |
| description | O | Описание операции |
| currency | O | Трехбуквенный код валюты (ISO 4217), например: RUB |
| amount | O | Сумма операции в минорных единицах. |
| type | R | Тип ввода параметров карты (card/card\_page). card - все необходимые атрибуты карты клиент указывает прямо в запросе. card\_page - никакие атрибуты карты в запросе указываться не должны. В ответ на запрос старта верификации Сервис вернёт сообщение со статусом redirect. В результате клиент будет перенаправлен на страницу Сервиса, на которой он должен будет ввести атрибуты карты |
| cardId | C | Идентификатор сохраненной ранее карты. Допустимая длина: 1-32 символа. Указывается при type «card», номер карты при этом не указывается. При type «card\_page» поле должно отсутствовать |
| pan | C | Номер карты. Указывается при type «card».  При type «card\_page» поле должно отсутствовать |
| expiry | C | Срок действия карты Указывается при type «card». При type «card\_page» поле должно отсутствовать |
| csc | C | Card Security Code (CVV2/CVC2) Формат: три цифры. Указывается при type «card».  При type «card\_page» поле должно отсутствовать |
| cardholder | O | Имя владельца карты. Допустимая длина: 1-26 символов Указывается при type «card».  При type «card\_page» поле должно отсутствовать |
| params.\* | O | Параметры операции (набор и формат параметров определяется магазином). |
| returnUrl | O | Адрес возврата для продолжения верификации после перенаправления. Обязательный параметр при поддержке перенаправлений. |
| state.\* | O | Опциональные параметры, позволяющие приложению использовать специфические режимы обработки отдельных состояний верификации карты. |
| submerchantName | C | Название подмагазина.  Поле обязательное для магазинов-посредников. |
| deviceFingerprint | O | Цифровой отпечаток устройства пользователя.  Например, хэш набора характеристик браузера клиента. Передается в систему фрод-мониторинга. |
| 3ds2.supported | O | Признак поддержки приложением протокола 3DS 2.  Возможные значения: true/false. По умолчанию false. |
| 3ds2.destWalletNum, 3ds2.destCardNum,  3ds2.destPhoneNum,  3ds2.destAcctNum | C | Описание получателя денежных средств при переводах на электронный кошелёк, карту, телефон или счёт.  Параметр обязательный при проведении 3DS v2 по картам МИР для некоторых категорий магазинов |
| lang | O | Язык операции (ISO 639-1), например, ru |

В ответ на запрос старта верификации карты Сервис может вернуть сообщение об ошибке или сообщение с состоянием транзакции верификации карты. В зависимости от значения поля state в сообщении с состоянием транзакции, приложение выполняет различные действия для продолжения верификации карты, приведенные в Таблица 39.

Таблица Значения поля state

| Значение поля “state” | Действие приложения |
| --- | --- |
| in\_progress | Через некоторое время выполнить новый запрос состояния. |
| **confirmation** | Запросить у клиента проверочную информацию. |
| redirect | Выполнить перенаправление на указанный адрес. |
| 3ds2\_prepare | Выбрать конкретную версию протокола 3DS 2 и собрать необходимые данные о клиенте и КИ |
| 3ds2\_challenge | Начать взаимодействие с ACS для проведения явной аутентификации клиента |
| 3ds2\_info | Отобразить клиенту информационное сообщение, связанное с прохождением аутентификации |
| iframe | Отобразить на web-странице фрейм с указанными параметрами |
| result | Отобразить информацию о результате верификации. |

## Обработка верификации

Правила работы с состоянием in\_progress совпадают с правилами, описанными для платежа в пункте 3.3.2.

## Подтверждение владения картой

Сценарий верификации может предполагать необходимость ввода клиентом проверочной информации. Например, это может быть ввод контрольной суммы, которую Сервис зарезервировал на карте. При получении от Сервиса состояния **confirmation**, приложение должно предложить пользователю ввести проверочные данные и отправить их Сервису.

В ответе с состоянием Сервис возвращает данные, приведенные в Таблица 40.

Таблица Данные для подтверждения

| Параметр |  | Описание |
| --- | --- | --- |
| **attemptsLeft** | R | Количество оставшихся попыток ввода проверочной информации |
| **failedAttempts** | R | Количество неуспешных попыток ввода проверочной информации |

Пример ответа:

HTTP/1.1 200 OK

Content-Type: application/json

Content-Length: 429

Expires: Thu, 01 Dec 1994 16:00:00 GMT

Cache-Control: no-cache

{

"state":"confirmation",

"failedAttempts":1,

"attemptsLeft":2

}

Для подтверждения владения картой приложение отправляет запрос:

POST /api/v4/<portal\_id>/card/verification/<token>/confirm

Параметры запроса приведены в Таблица 41.

Таблица Параметры запроса

|  |  |  |
| --- | --- | --- |
| Параметр |  | Описание |
| amount | R | Контрольная сумма в минорных единицах. |

В ответ Сервис возвращает JSON-структуру с информацией для продолжения верификации карты.

В случае отказа от ввода проверочных данных приложение отправляет запрос на прекращение верификации карты:

POST /api/v4/<portal\_id>/card/verification/<token>/cancel

## Перенаправления

Правила работы с состоянием redirect совпадают с правилами, описанными для платежа в пункте 3.6.

## Получение текущего состояния и продолжение верификации карты

В случае получения в ответ от Сервиса сообщения с состоянием in\_progress или возврата по адресу, указанному в параметре returnUrl после перенаправления, приложение должно продолжить выполнение верификации карты, выполнив запрос на получение текущего состояния верификации:

POST /api/v4/<portal\_id>/card/verification/<token>

В ответном сообщении Сервис возвращает JSON-структуру с информацией о текущем состоянии верификации. В зависимости от значения поля state в ответе Сервиса приложение выполняет соответствующее действие для продолжения верификации.

Веб-приложение может использовать скрипт state.js (см. [скрипт state.js](#_Скрипт_state.js)) для упрощения обработки состояния in\_progress.

## Выполнение аутентификации по протоколу 3DS v2

Правила выполнения аутентификации при верификации карты совпадают с правилами, описанными для платежа в пункте 3.11.

## Отображение результата верификации карты

При получении от Сервиса состояния result приложение отображает страницу с результатом верификации. В ответе Сервиса содержатся параметры, приведенные в Таблица 42.

Расширенные коды завершения верификации карты показаны в Приложение 3. Расширенные коды завершения верификации или регистрации карты.

Таблица Описание результата верификации

| Параметр |  | Описание |
| --- | --- | --- |
| token | R | Токен, уникальный идентификатор верификации. |
| startedAt | R | Время начала операции (Unix time в ms) |
| finishedAt | R | Время завершения операции (Unix time в ms) |
| result/\* | R | Детали результата верификации |

Параметры результата платежа (result) приведены в Таблица 43.

Таблица Параметры результата платежа

| Параметр |  | Описание |
| --- | --- | --- |
| status | R | Статус транзакции:   * SUCCESS - транзакция завершена успешно; * DECLINED - клиент отказался от продолжения транзакции; * FAILED - транзакция завершена неуспешно; * UNKNOWN - статус транзакции не определен. |
| extendedCode | R | Расширенный код завершения транзакции (см. Приложение 2. Расширенные коды завершения платежа). |
| responseCode | O | Код ответа процессинга. |
| amountReverted | C | Признак успешного отката (возврата) зарезервированных сумм |
| trxId | O | Идентификатор транзакции, совпадает с токеном |
| orderStatus | O | Статус заказа.  Возможные значения:   * unknown - статус заказа неизвестен (или нет дополнительного статуса у данного заказа); * registered - заказ зарегистрирован у продавца (в магазине); * registration\_attempts\_exhausted - исчерпаны попытки регистрации заказа у продавца (в магазине); |
| 3ds | O | Результата проведения 3DS-аутентификации. Заполняется в некоторых сценариях. |
| 3ds/status | R | Cтатус результата проведения 3DS-аутентификации. Возможные значения указаны в приложении 4. |
| 3ds/eci | O | Electronic Commerce Indicator. Индикатор, показывающий уровень безопасности транзакции. Значение устанавливается в соответствие с результатом аутентификации. |
| 3ds/cavv | O | Cardholder Authentication Verification Value. Контрольное значение, вычисляемое средствами процессингового центра, необходимое для проверки на стадии авторизации. |
| 3ds/xid | O | Идентификатор транзакции MPI. |
| 3ds/dsTransID | O | Идентификатор транзакции Directory Server |

Пример ответа:

HTTP/1.1 200 OK

Content-Type: application/json

Content-Length: 592

Expires: Thu, 01 Dec 1994 16:00:00 GMT

Cache-Control: no-cache

{

token: "1234567890ABCDEF",

startedAt: "5555555555111",

finishedAt: "5555555555222",

state: "result",

result: {

status: "SUCCESS",

extendedCode: "OK",

trxId: "1234567890ABCDEF",

3ds: {

cavv: "123456123456123456",

xid: "123456712345671234567",

eci: "05",

status: "PARES\_Y"

}

}

}

# Регистрация карт

Регистрация карты – это сохранение атрибутов карты в БД Сервиса с привязкой к магазину. Регистрация карты выполняется только после её успешной верификации.  
После того, как карта зарегистрирована, её можно использовать в платежах без повторного ввода атрибутов карты и в рекуррентных платежах.

## Сценарий взаимодействия

Сценарий регистрации карты с точки зрения последовательности запросов полностью совпадает со сценарием верификации карты.

Типы запросов, используемые при регистрации, приведены в Таблица 44.

Таблица Типы запросов

| Запрос | Описание | Результат |
| --- | --- | --- |
| POST /api/v4/<portal\_id>/token | Получение токена. | token |
| POST /api/v4/<portal\_id>/card/registration/<token>/start | Старт регистрации. | state |
| POST /api/v4/<portal\_id>/card/registration/<token>/confirm | Подтверждение регистрации | state |
| POST /api/v4/<portal\_id>/card/registration/<token>/cancel | Отказ от завершения регистрации | state |
| POST /api/v4/<portal\_id>/ card/ registration /<token>/3ds2-prepare/accept | Подтверждение подготовки к проведению аутентификации по протоколу 3DS 2 | state |
| POST /api/v4/<portal\_id>/ card/ registration /<token>/3ds2-prepare/reject | Отказ от проведения аутентификации по протоколу 3DS 2 | state |
| POST /api/v4/<portal\_id>/card/registration/<token> | Получение состояния операции | state |

Сценарий регистрации карты представлен на Рисунок 5.



Рисунок 5 Сценарий регистрации карты

Обозначения на диаграмме последовательности:

* Opt – опциональный блок. Присутствует при выполнении некоторого условия;
* Alt – альтернативный сценарий. Подразумевает выполнение только одного из предложенных вариантов;
* Loop – повторение действия. Подразумевает повторное выполнение действия, пока не будет выполнено некоторое условие.

Прямой поток событий сценария регистрации карты:

1. Клиент инициирует старт регистрации карты.
   1. UI запрашивает в PGA получение токена.
   2. UI отправляет запрос в PGA на старт регистрации карты.
   3. Опционально. При получении ответа со статусом “in\_progress” UI отправляет запрос в PGA на получение статуса регистрации.
   4. Опционально. При получении ответа со статусом “confirm” UI отображает клиенту страницу для ввода проверочных данных.
   5. Опционально. Клиент вводит проверочные данные.
   6. Опционально. UI отправляет запрос в PGA на подтверждение владения картой.
   7. Опционально. Клиент отказывается от ввода проверочных данных.
   8. Опционально. UI отправляет запрос в PGA на прекращение регистрации карты.
   9. Опционально. При получении ответа со статусом “in\_progress” UI отправляет запрос в PGA на получение статуса регистрации.
   10. Опционально. При получении ответа со статусом “redirect” UI переадресовывает клиента на страницу ACS банка-эмитента или на страницу ввода параметров карты.
   11. Опционально. UI отправляет запрос в PGA на получение статуса регистрации.
   12. Опционально. При получении ответа со статусом “in\_progress” UI отправляет запрос в PGA на получение статуса регистрации.
   13. PGA сохраняет атрибуты карты в БД.
   14. PGA передает в UI результат регистрации карты.
   15. UI отображает клиенту результат регистрации карты.

## Получение токена транзакции

Перед началом каждой регистрации приложение получает уникальный токен транзакции регистрации карты, который используется для идентификации транзакции во всех последующих запросах, а также служит для исключения дубликатов при повторе запросов.

Формат запроса и ответа полностью совпадает с форматом, описанным в пункте 3.2.

## Старт регистрации карты

Запрос на старт регистрации карты:

POST /api/v4/<portal\_id>/card/registration/<token>/start

где <token> – значение токена, полученное на предыдущем шаге.

Параметры запроса такие же, как при старте верификации карты (п.4.3). И есть дополнительные параметры (Таблица 45).

Таблица Дополнительные параметры запроса

|  |  |  |
| --- | --- | --- |
| Параметр |  | Описание |
| title | O | Название карты. Если поле не указано, то имя сгенерирует Сервис |
| externalToken | C | Поле передается при регистрации виртуальной карты, т.е. если operationId=virtual |

В зависимости от значения поля state в ответе Сервиса, приложение выполняет различные действия для продолжения регистрации карты, приведенные в Таблица 46.

Таблица Значения поля state

| Значение поля “state” | Действие приложения |
| --- | --- |
| in\_progress | Через некоторое время выполнить новый запрос состояния. |
| **confirmation** | Запросить у клиента проверочную информацию. |
| redirect | Выполнить перенаправление на указанный адрес. |
| 3ds2\_prepare | Выбрать конкретную версию протокола 3DS 2 и собрать необходимые данные о клиенте и КИ |
| 3ds2\_challenge | Начать взаимодействие с ACS для проведения явной аутентификации клиента |
| 3ds2\_info | Отобразить клиенту информационное сообщение, связанное с прохождением аутентификации |
| iframe | Отобразить на web-странице фрейм с указанными параметрами |
| result | Отобразить информацию о результате регистрации. |

## Обработка регистрации

Правила работы с состоянием in\_progress совпадают с правилами, описанными для платежа в пункте 3.3.2.

## Подтверждение владения картой

Правила работы с состоянием **confirmation** совпадают с правилами, описанными для верификации карты в пункте 4.5.

## Перенаправления

Правила работы с состоянием redirect совпадают с правилами, описанными для платежа в пункте 3.6.

## Получение текущего состояния и продолжение регистрации карты

В случае получения в ответ от Сервиса сообщения с состоянием in\_progress или возврата по адресу, указанному в параметре returnUrl после перенаправления, приложение должно продолжить выполнение верификации карты, выполнив запрос на получение текущего состояния верификации:

POST /api/v4/<portal\_id>/card/verification/<token>

В ответном сообщении Сервис возвращает JSON-структуру с информацией о текущем состоянии верификации. В зависимости от значения поля state в ответе Сервиса приложение выполняет соответствующее действие для продолжения верификации.

Веб-приложение может использовать скрипт state.js (см. [скрипт state.js](#_Скрипт_state.js)) для упрощения обработки состояния in\_progress.

## Выполнение аутентификации по протоколу 3DS v2

Правила выполнения аутентификации при регистрации карты совпадают с правилами, описанными для платежа в пункте 3.11.

## Отображение результата регистрации карты

При получении от Сервиса состояния result приложение отображает страницу с результатом регистрации. Набор параметров в ответе Сервиса такой же, как и при верификации карты. Дополнительно в параметре result при успешной регистрации карты содержится идентификатор карты. В дальнейшем по этому идентификатору можно ссылаться на карту при проведении платежей (Таблица 47).

Расширенные коды завершения регистрации карты показаны в Приложение 3. Расширенные коды завершения верификации или регистрации карты.

Таблица Дополнительные параметры результата платежа

| Параметр |  | Описание |
| --- | --- | --- |
| paymentSystem | C | Идентификатор платёжной системы карты. Возможные значения:   * VISA, * MASTERCARD, * MIR, * AMEX, * UNIONPAY, * GCLOCAL |
| cardId | C | Идентификатор зарегистрированной карты |
| maskedPan | О | Маскированный PAN карты |

Пример ответа:

HTTP/1.1 200 OK

Content-Type: application/json

Content-Length: 592

Expires: Thu, 01 Dec 1994 16:00:00 GMT

Cache-Control: no-cache

{

token: "1234567890ABCDEF",

startedAt: "5555555555111",

finishedAt: "5555555555222",

state: "result",

result: {

status: "SUCCESS",

extendedCode: "OK",

trxId: "1234567890ABCDEF",

responseCode: "000",  
 orderStatus: "registered",  
 cardId: "12AAABBBCCC",  
 maskedPan: "554321\*\*\*\*\*\*\*\*3244",

paymentSystem: “VISA”

}

}

# Протокол MirPayAPI (устаревший протокол)

* Начиная с версии PGA 4.2.86, рекомендуется проводить платёж по протоколу OpenAPI. Сценарий взаимодействия описан в пункте 3.8 «Подтверждение платежа с использованием Mir Pay».

Для поддержки платежей MirPay в PGA реализован протокол MirPayAPI, который используется, для:

* получения данных In-Application операции (JWE) от мобильного приложения MirPay для проведения In-Application операции, по запросу: PUT /mirpay/<token> (Таблица 48);
* передачи JWKS по запросу: GET /mirpay/jwks

Пример получения данных In-Application операции JWE (запрос/ответ):

PUT /mirpay/<token> HTTP/1.1

{

"orderId": <token>,

"cryptogram": <cryptogram>"  
 }

HTTP/1.1 202 Accepted

Таблица 48 Параметры запроса PUT /mirpay/<token>

| Параметр |  | Описание |
| --- | --- | --- |
| orderId | R | Идентификатор заказа/транзакции ТСП |
| cryptogram | R | Криптограмма с данными In-Application платежа в формате JWE Compact Serialization |

Пример передачи JWKS (запрос/ответ):

GET /mirpay/jwks HTTP/1.1

HTTP/1.1 200 OK

Content-Type: application/json; charset=UTF-8

{

"keys": [{

"kid": <X.509 certificate serial number, number in hex format("5938a…ef63c9")>,

"kty": "RSA",

"use": "sig",

"x5c": [

<end point merchant sertificate>,

<Acquirer SubCA (or Concentrator SubCA)>

]

}]

}

# Протокол «Данные карт Gazprom Pay»

Данный протокол работает только в локальной сети. В ответе PGA передает JSON с полным PAN и сроком действия карты. Взаимодействие осуществляется по протоколу HTTPS c использованием метода POST в кодировке UTF-8. Для всех запросов используется базовый адрес:

https://<host>:<port>/gazprompayCardData/<portalId>/<cardId>

где

* <host> и <port> - адрес сервера;
* <portalId> – идентификатор портала;
* <cardId> - идентификатор банковской карты.

В ответ от PGA SVFE получит JSON-структуру с данными карты для данного cardId:

HTTP/1.1 200 OK

Content-Type: application/json

Content-Length: 51

Expires: Thu, 01 Dec 2022 16:00:00 GMT

Cache-Control: no-cache

{

"pan":"panValue",

"expiry":"MMYY"

"paymentSystem":"MASTERCARD"

}

Начиная с версии PGA 4.2.79 в ответе на запрос добавлено поле "paymentSystem". Возможные значения: VISA, MASTERCARD, MIR, UNIONPAY.

Возможные ошибки:

1. "Карта не найдена" (HTTP/1.1 404 Card not found) – в случаях, когда карта не найдена, заблокирована или скрыта.
2. "Запрещено настройками" (HTTP/1.1 400 Restricted by settings) – если портал заблокирован или не найден связанный с ним партнер.
3. Ошибки обработки запроса (HTTP/1.1 400 Invalid request url) – в случае некорректного адреса запроса.

Пример передачи данных карт (запрос/ответ):

POST 11.11.11.11:22222/gazprompayCardData/ACQPORTAL/A1RXJKI00533

{

"pan":"5454545454545454",

"expiry":"1223"

"paymentSystem":"MASTERCARD"

}

# Примеры последовательностей сообщений

Для удобства работы с примерами параметры в теле запросов размещены на отдельных строках без символа «&» и без URL-кодирования.

Тела сообщений других протоколов в примерах не указаны.

## Оплата покупки банковской картой, 3DS v1, обычный режим возврата

// Запрос клиентским приложением нового токена

POST /api/v4/TESTPORTAL123/token HTTP/1.1

// Ответ с токеном

HTTP/1.1 200 OK

Content-Type: application/json

{"token":"1234567890ABCDEF"}

// Запрос клиентским приложением старта платежа

POST /api/v4/TESTPORTAL123/payment/1234567890ABCDEF/start HTTP/1.1

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

merchantId=test\_merchant

merchantTrx=trx-12345

accountId=test\_account

src.type=card

src.pan=5000000000000000

src.expiry=1030

src.csc=123

amount=123456

currency=RUB

description=Sample

returnUrl=**https://site.ru/c2c?id=1234567890ABCDEF**

state.redirect=post\_params

// Ответ Сервиса с запросом подтверждения оферты

HTTP/1.1 200 OK

Content-Type: application/json

{

"token": "1234567890ABCDEF"

"state": "offer",

"description": "Sample",

"descriptionParams": {  
 "acquirerCommission":"300",  
 "amountWithAcquirerCommission":"123756"  
 },

"merchant": {

"name": "Test Merchant",

"logoUrl": "https://testmerchant.ru/logo.png",

"url": <https://testmerchant.ru>,

“emailReceiptEnabled”: true

},

"src":{

"type":"card",

"pan":"500000xxxxxx0000",

“paymentSystem”: “MASTERCARD”

},

"amount":"123456",

"currency":"RUB",

"providerAlias": "gcs"

}

// Клиент согласился с офертой

POST /api/v4/TESTPORTAL123/payment/1234567890ABCDEF/accept HTTP/1.1

// Сервис предлагает выполнить переход для проведения 3DS

HTTP/1.1 200 OK

Content-Type: application/json

{

"state": "redirect",

"url": "**acs.issuer.com/3ds**",

"post": {

"PaReq": "IqjKNG1CbhBsjtVYuo6e9cYI7I7tdi90OS93LgX25lrxU=",

"MD": "87654321",

"TermUrl": "**https://service.ru/api/v4/TESTPORTAL123/payment/1234567890ABCDEF/return**"

}

}

// Браузер клиента переходит по адресу **acs.issuer.com/3ds**

// Клиент проходит 3DS

// Результат этого прохождения возвращается Сервису

// Сервис "возвращает" клиента клиентскому приложению

HTTP/1.1 302 Moved Temporarily

Location: **https://site.ru/c2c?id=1234567890ABCDEF**

// Клиентское приложение запрашивает текущий статус транзакции

GET /api/v4/TESTPORTAL123/payment/1234567890ABCDEF HTTP/1.1

// Сервис возвращает статус «Платёж в обработке»

HTTP/1.1 200 OK

Content-Type: application/json

{

"state": "in\_progress"

}

// Через некоторое время клиентское приложение снова запрашивает текущий статус транзакции

GET /api/v4/TESTPORTAL123/payment/1234567890ABCDEF HTTP/1.1

// Сервис возвращает финальный статус

HTTP/1.1 200 OK

Content-Type: application/json

{

"token": "1234567890ABCDEF",

"startedAt" : "1456407089068",

"finishedAt" : "1456407089406",

"state": "result",

"description": "Sample",

"merchant": {

"name": "Test Merchant",

"logoUrl": "https://testmerchant.ru/logo.png",

"url": <https://testmerchant.ru>,

“emailReceiptEnabled”: true

},

"src": {

"type": "card",

"pan": "500000xxxxxx0000",

“paymentSystem”: “MASTERCARD”

},

"amount": "123456",

"currency": "RUB",

"result": {

"status": "SUCCESS",

"extendedCode": "OK",

"trxId": "1234567890",

"responseCode": "00",

"rrn": "501617048902",

"approvalCode": "437762",

"isFullyAuthenticated": "true",

"authorizationTime": "0110122234"

},

"portalType": "web",

"type": "ACQUIRING"

}

## Оплата покупки банковской картой, 3DS v1, прямой режим возвратов

// Запрос клиентским приложением нового токена

POST /api/v4/TESTPORTAL123/token HTTP/1.1

// Ответ с токеном

HTTP/1.1 200 OK

Content-Type: application/json

{"token":"1234567890ABCDEF"}

// Запрос клиентским приложением старта платежа

POST /api/v4/TESTPORTAL123/payment/1234567890ABCDEF/start HTTP/1.1

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

merchantId=test\_merchant

merchantTrx=trx-12345

accountId=test\_account

src.type=card

src.pan=5000000000000000

src.expiry=1030

src.csc=123

amount=123456

currency=RUB

description=Sample

returnUrl=**https://site.ru/c2c?id=1234567890ABCDEF**

state.redirect=post\_params

// Ответ Сервиса с запросом подтверждения оферты

HTTP/1.1 200 OK

Content-Type: application/json

{

"token": "1234567890ABCDEF"

"state": "offer",

"description": "Sample",

"descriptionParams": {  
 "acquirerCommission":"300",  
 "amountWithAcquirerCommission":"123756"  
 },

"merchant": {

"name": "Test Merchant",

"logoUrl": "https://testmerchant.ru/logo.png",

"url": https://testmerchant.ru,

“emailReceiptEnabled”: true

},

"src":{

"type":"card",

"pan":"500000xxxxxx0000",

“paymentSystem”: “MASTERCARD”

},

"amount":"123456",

"currency":"RUB",

"providerAlias": "gcs"

}

// Клиент согласился с офертой

POST /api/v4/TESTPORTAL123/payment/1234567890ABCDEF/accept HTTP/1.1

// Сервис предлагает выполнить переход для проведения 3DS

HTTP/1.1 200 OK

Content-Type: application/json

{

"state": "redirect",

"url": "**acs.issuer.com/3ds**",

"post": {

"PaReq": "IqjKNG1CbhBsjtVYuo6e9cYI7I7tdi90OS93LgX25lrxU=",

"MD": "87654321",

"TermUrl": "**https://site.ru/c2c?id=1234567890ABCDEF**"

}

}

// Браузер клиента переходит по адресу **acs.issuer.com/3ds**

// Клиент проходит 3DS

// Результат этого прохождения возвращается напрямую в клиентское приложение на

// адрес **https://site.ru/c2c?id=1234567890ABCDEF**

// Клиентское приложение сообщает Сервису результат прохождения 3DS

POST /api/v4/TESTPORTAL123/payment/1234567890ABCDEF/merchant-return HTTP/1.1

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

PaRes=3874874KNG1CbhBsjtVYuo6e9cYI7I7tdi90OS93LgX25lru=

MD=234234234

// Сервис возвращает статус «Платёж в обработке»

HTTP/1.1 200 OK

Content-Type: application/json

{

"state": "in\_progress"

}

// Через некоторое время клиентское приложение снова запрашивает текущий

// статус транзакции

GET /api/v4/TESTPORTAL123/payment/1234567890ABCDEF HTTP/1.1

// Сервис возвращает финальный статус

HTTP/1.1 200 OK

Content-Type: application/json

{

"token": "1234567890ABCDEF",

"startedAt" : "1456407089068",

"finishedAt" : "1456407089406",

"state": "result",

"description": "Sample",

"merchant": {

"name": "Test Merchant",

"logoUrl": "https://testmerchant.ru/logo.png",

"url": https://testmerchant.ru,

“emailReceiptEnabled”: true

},

"src": {

"type": "card",

"pan": "500000xxxxxx0000",

“paymentSystem”: “MASTERCARD”

},

"amount": "123456",

"currency": "RUB",

"result": {

"status": "SUCCESS",

"extendedCode": "OK",

"trxId": "1234567890",

"responseCode": "00",

"rrn": "501617048902",

"approvalCode": "437762",

"isFullyAuthenticated": "true",

"authorizationTime": "0110122234"

},

"portalType": "web",

"type": "ACQUIRING"

}

## Оплата покупки банковской картой, 3DS v2.1.0, мобильное приложение

// Запрос клиентским приложением нового токена

POST /api/v4/TESTPORTAL123/token HTTP/1.1

// Ответ с токеном

HTTP/1.1 200 OK

Content-Type: application/json

{"token":"1234567890ABCDEF"}

// Запрос клиентским приложением старта платежа

POST /api/v4/TESTPORTAL123/payment/1234567890ABCDEF/start HTTP/1.1

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

merchantId=test\_merchant

merchantTrx=trx-12345

accountId=test\_account

src.type=card

src.pan=5000000000000000

src.expiry=1030

src.csc=123

amount=123456

currency=RUB

description=Sample

returnUrl=https://site.ru/c2c?id=1234567890ABCDEF

state.redirect=post\_params

3ds2.supported=true

// Ответ Сервиса с запросом подтверждения оферты

HTTP/1.1 200 OK

Content-Type: application/json

{

"token": "1234567890ABCDEF"

"state": "offer",

"description": "Sample",

"descriptionParams": {  
 "acquirerCommission":"300",  
 "amountWithAcquirerCommission":"123756"  
 },

"merchant": {

"name": "Test Merchant",

"logoUrl": "https://testmerchant.ru/logo.png",

"url": https://testmerchant.ru,

“emailReceiptEnabled”: true

},

"src":{

"type":"card",

"pan":"500000xxxxxx0000",

“paymentSystem”: “MASTERCARD”

},

"amount":"123456",

"currency":"RUB",

"providerAlias": "gcs"

}

// Клиент согласился с офертой

POST /api/v4/TESTPORTAL123/payment/1234567890ABCDEF/accept HTTP/1.1

// Сервис начинает подготовку к 3DS 2

HTTP/1.1 200 OK

Content-Type: application/json

{

state: "3ds2\_prepare",  
 versions: ["2.1.0", “2.2.0”],  
 directoryServerID: "TESTDS"

}

// МП передаёт в SDK версию протокола и идентификатор DS. В ответ получает

// параметры проведения аутентификации

POST /api/v4/TESTPORTAL123/payment/1234567890ABCDEF/3ds2-prepare/accept HTTP/1.1  
Content-Type: application/x-www-form-urlencoded  
  
version=2.1.0  
device.channel=APP  
device.sdkAppID=12345678-1234-1234-123456789012  
device.sdkEncData=ABCD123…EFF123  
device.sdkEphemPubKey={"kty":"EC","crv":"P-256","x": "1234567890123456789012345678901234567890123","y":"123456789012345678901234567890\_1234567890123"}  
device.sdkMaxTimeout=10  
device.sdkReferenceNumber=3DS\_LOA\_SDK\_1234\_123456\_12345  
device.sdkTransID=12345678-1234-1234-123456789012  
device.sdkInterface=01  
device.sdkUiType=01,02,03,04

// Сервис отправляет в DS запрос AReq и получает в ответ ARes со статусом C

// Сервис сообщает МП о начале явной аутентификации

HTTP/1.1 200 OK

Content-Type: application/json

{

state: "3ds2\_challenge",  
 "3dsServerTransID":"12345678-1234-1234-123456789012",  
 "acsTransID": "12345678-1234-1234-123456789012",  
 "acsReferenceNumber": "ACS",  
 "acsSignedContent": "ABCD2323…DEFF345"

}

// МП передаёт полученные данные в SDK и начинается взаимодействие МП и ACS

// После завершения этого взаимодействия МП запрашивает текущий статус транзакции

GET /api/v4/TESTPORTAL123/payment/1234567890ABCDEF HTTP/1.1

// Сервис возвращает статус «Платёж в обработке»

HTTP/1.1 200 OK

Content-Type: application/json

{

"state": "in\_progress"

}

// Через некоторое время МП снова запрашивает текущий статус транзакции

GET /api/v4/TESTPORTAL123/payment/1234567890ABCDEF HTTP/1.1

// Сервис возвращает финальный статус

HTTP/1.1 200 OK

Content-Type: application/json

{

"token": "1234567890ABCDEF",

"startedAt" : "1456407089068",

"finishedAt" : "1456407089406",

"state": "result",

"description": "Sample",

"merchant": {

"name": "Test Merchant",

"logoUrl": "https://testmerchant.ru/logo.png",

"url": https://testmerchant.ru,

“emailReceiptEnabled”: true

},

"src": {

"type": "card",

"pan": "500000xxxxxx0000",

“paymentSystem”: “MASTERCARD”

},

"amount": "123456",

"currency": "RUB",

"result": {

"status": "SUCCESS",

"extendedCode": "OK",

"trxId": "1234567890",

"responseCode": "00",

"rrn": "501617048902",

"approvalCode": "437762",

"orderStatus": "REGISTERED",  
 "orderStatusChangedAt": "1456407089068"

},

"portalType": "MOBILE",

"type": "ACQUIRING"

}

## Оплата покупки банковской картой, 3DS v2.1.0, Web-приложение, обычный режим возврата

// Запрос клиентским приложением нового токена

POST /api/v4/TESTPORTAL123/token HTTP/1.1

// Ответ с токеном

HTTP/1.1 200 OK

Content-Type: application/json

{"token":"1234567890ABCDEF"}

// Запрос клиентским приложением старта платежа

POST /api/v4/TESTPORTAL123/payment/1234567890ABCDEF/start HTTP/1.1

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

merchantId=test\_merchant

merchantTrx=trx-12345

accountId=test\_account

src.type=card

src.pan=5000000000000000

src.expiry=1030

src.csc=123

amount=123456

currency=RUB

description=Sample

returnUrl=**https://site.ru/c2c?id=1234567890ABCDEF**

state.redirect=post\_params

3ds2.supported=true

// Ответ Сервиса с запросом подтверждения оферты

HTTP/1.1 200 OK

Content-Type: application/json

{

"token": "1234567890ABCDEF"

"state": "offer",

"description": "Sample",

"descriptionParams": {  
 "acquirerCommission":"300",  
 "amountWithAcquirerCommission":"123756"  
 },

"merchant": {

"name": "Test Merchant",

"logoUrl": "https://testmerchant.ru/logo.png",

"url": https://testmerchant.ru,

“emailReceiptEnabled”: true

},

"src":{

"type":"card",

"pan":"500000xxxxxx0000",

“paymentSystem”: “MASTERCARD”

},

"amount":"123456",

"currency":"RUB",

"providerAlias": "gcs"

}

// Клиент согласился с офертой

POST /api/v4/TESTPORTAL123/payment/1234567890ABCDEF/accept HTTP/1.1

// Сервис начинает подготовку к 3DS 2

HTTP/1.1 200 OK

Content-Type: application/json

{

state: "3ds2\_prepare",  
 versions: ["2.1.0", “2.2.0”],  
 directoryServerID: "TESTDS"

}

// Web-приложение передаёт сервису параметры браузера

POST /api/v4/TESTPORTAL123/payment/1234567890ABCDEF/3ds2-prepare/accept HTTP/1.1  
Content-Type: application/x-www-form-urlencoded  
  
version=2.1.0  
device.channel=BRW

device.browserAcceptHeader=text/html  
device.browserIP=127.0.0.1  
device.browserJavaEnabled=false  
device.browserLanguage=RU  
device.browserColorDepth=32  
device.browserScreenHeight=800  
device.browserScreenWidth=480  
device.browserTZ=180  
device.browserUserAgent=Gecko  
device.challengeWindowSize=02  
iframeReturnUrl=**https://testmerchant.ru/iframe.php**

// Если бы в этом сообщении не было параметра iframeReturnUrl, то в качестве адреса возврата после iframe-перенаправления Сервис использовал бы адрес [**https://site.ru/c2c?id=1234567890ABCDEF**](https://site.ru/c2c?id=1234567890ABCDEF)из запроса старта платежа

// Сервис отправляет в DS запрос AReq. В поле AReq.notificationUrl Сервис указывает свой адрес возврата

// Сервис получает в ответ ARes со статусом C

// Сервис сообщает Web-приложению о начале явной аутентификации

HTTP/1.1 200 OK

Content-Type: application/json

{

"state": "iframe",  
 "iframeWidth": 390,  
 "iframeHeight": 400,  
 "url": "**test-acs.com\3ds-2**",  
 "post":{  
 "creq": "ABCD123…DEF234=="  
 }

}

// Web-приложение создаёт фрейм и начинается взаимодействие клиента и ACS

// После завершения этого взаимодействия ACS отправляет Сервису сообщение RReq с результатом аутентификации

// Сервис отвечает сообщением RRes

// ACS перенаправляет клиента на адрес возврата Сервиса. В запросе ACS указывает HTTP-параметр cres.

// Сервис получает запрос на свой адрес возврата. В запросе есть HTTP-параметр cres, но он не несёт полезной информации, потому что Сервис уже получил сообщение RReq.

// Сервис "возвращает" клиента Web-приложению

HTTP/1.1 302 Moved Temporarily

Location: **https://testmerchant.ru/iframe.php**

Web-приложение запрашивает текущий статус транзакции

GET /api/v4/TESTPORTAL123/payment/1234567890ABCDEF HTTP/1.1

// Сервис возвращает статус «Платёж в обработке»

HTTP/1.1 200 OK

Content-Type: application/json

{

"state": "in\_progress»

}

// Через некоторое время Web-приложение снова запрашивает текущий статус транзакции

GET /api/v4/TESTPORTAL123/payment/1234567890ABCDEF HTTP/1.1

// Сервис возвращает финальный статус

HTTP/1.1 200 OK

Content-Type: application/json

{

“token”: “1234567890ABCDEF”,

“startedAt” : “1456407089068”,

“finishedAt” : “1456407089406”,

“state”: “result”,

“description”: “Sample”,

“merchant”: {

“name”: “Test Merchant”,

“logoUrl”: “https://testmerchant.ru/logo.png”,

“url”: <https://testmerchant.ru>,

“emailReceiptEnabled”: true

},

“src”: {

“type”: “card”,

“pan”: “500000xxxxxx0000”,

“paymentSystem”: “MASTERCARD”

},

“amount”: “123456”,

“currency”: “RUB”,

“result”: {

“status”: “SUCCESS”,

“extendedCode”: “OK”,

“trxId”: “1234567890”,

“responseCode”: “00”,

“rrn”: “501617048902”,

“approvalCode”: “437762”,

“orderStatus”: “REGISTERED”,  
 “orderStatusChangedAt”: “1456407089068”

},

“portalType”: “MOBILE”,

“type”: “ACQUIRING”

}

## Оплата покупки банковской картой, 3DS v2.2.0, Web-приложение, прямой режим возврата, выполнение 3DS-метода

// Запрос клиентским приложением нового токена

POST /api/v4/TESTPORTAL123/token HTTP/1.1

// Ответ с токеном

HTTP/1.1 200 OK

Content-Type: application/json

{«token»:»1234567890ABCDEF»}

// Запрос клиентским приложением старта платежа

POST /api/v4/TESTPORTAL123/payment/1234567890ABCDEF/start HTTP/1.1

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

merchantId=test\_merchant

merchantTrx=trx-12345

accountId=test\_account

src.type=card

src.pan=5000000000000000

src.expiry=1030

src.csc=123

amount=123456

currency=RUB

description=Sample

returnUrl=**https://site.ru/c2c?id=1234567890ABCDEF**

state.redirect=post\_params

3ds2.supported=true

// Ответ Сервиса с запросом подтверждения оферты

HTTP/1.1 200 OK

Content-Type: application/json

{

“token”: “1234567890ABCDEF”

“state”: “offer”,

“description”: “Sample”,

“descriptionParams”: {  
 “acquirerCommission”:”300”,  
 “amountWithAcquirerCommission”:”123756”  
 },

“merchant”: {

“name”: “Test Merchant”,

“logoUrl”: “https://testmerchant.ru/logo.png”,

“url”: <https://testmerchant.ru>,

“emailReceiptEnabled”: true

},

“src”:{

“type”:”card”,

“pan”:”500000xxxxxx0000”,

“paymentSystem”: “MASTERCARD”

},

“amount”:”123456”,

“currency”:”RUB”,

“providerAlias”: “gcs”

}

// Клиент согласился с офертой

POST /api/v4/TESTPORTAL123/payment/1234567890ABCDEF/accept HTTP/1.1

// Сервис начинает подготовку к 3DS 2

HTTP/1.1 200 OK

Content-Type: application/json

{

state: “3ds2\_prepare”,  
 versions: [“2.1.0”, “2.2.0”],  
 directoryServerID: “TESTDS”

}

// Web-приложение передаёт сервису параметры браузера

POST /api/v4/TESTPORTAL123/payment/1234567890ABCDEF/3ds2-prepare/accept HTTP/1.1  
Content-Type: application/x-www-form-urlencoded  
  
version=2.2.0  
device.channel=BRW

device.browserAcceptHeader=text/html  
device.browserIP=127.0.0.1  
device.browserJavascriptEnabled=false  
device.browserLanguage=RU  
device.browserUserAgent=Gecko  
device.challengeWindowSize=02  
iframeReturnUrl=**https://testmerchant.ru/iframe.php**

// Если бы в этом сообщении не было параметра iframeReturnUrl, то в качестве адреса

// возврата после iframe-перенаправления Сервис использовал бы адрес

// [**https://site.ru/c2c?id=1234567890ABCDEF**](https://site.ru/c2c?id=1234567890ABCDEF)из запроса старта платежа

// Сервис определил, что в настройках Directory Server, через который нужно

// проводить 3DS, для диапазона карт, в который входит карта клиента, в поле

// threeDSMethodURL указано значение **https://test-acs.com\3ds-method**.

// Это значит, что нужно выполнить 3DS-метод

// Сервис отправляет в КИ команду на создание скрытого фрейма

HTTP/1.1 200 OK

Content-Type: application/json

{

“state”: “iframe”,  
 “url”: “**https://test-acs.com\3ds-method**”,  
 “post”:{  
 “threeDSMethodData”: “ABCD123…DEF234==”  
 }

}

// В этом сообщении в параметре threeDSMethodData содержится закодированный

// Json-документ такого вида:

//{  
// «threeDSMethodNotificationURL»:» **https://testmerchant.ru/iframe.php** «,

// «threeDSServerTransID»:»2342342534345345456746»

//}

// После отправки сообщения со статусом iframe Сервис 10 секунд ждёт запроса

// «Возврат клиента в магазин» на адрес /merchant-return

// Если время ожидания истекает, то Сервис считает, что 3DS-метод завершился

// неуспешно. Поэтому важно, чтобы при возврате Клиента из iframe-перенаправления

// Web-приложение отправило в Сервис запрос «Возврат клиента в магазин»

// Web-приложение создаёт скрытый фрейм, в результате чего происходит обращение

// из браузера Клиента на адрес ACS

// ACS собирает данные о браузере Клиента

// ACS перенаправляет клиента на адрес, указанный в параметре

// threeDSMethodNotificationURL, т.е. на адрес возврата Web-приложения.

// В параметр threeDSMethodData HTTP-запроса ACS помещает новый

// закодированный Json-документ, который содержит только одно поле:

//{  
// «threeDSServerTransID»:»2342342534345345456746»

//}

// Web-приложение закрывает скрытый фрейм и отправляет в Сервис сообщение о

// том, что Клиент вернулся после перенаправления

POST /api/v4/TESTPORTAL123/payment/1234567890ABCDEF/merchant-return HTTP/1.1

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

threeDSMethodData=121253543874874KbsjtVYuo6e9cYI3543547I7tdi90OS93LgX25lru=

// Сервис отправляет в DS запрос Areq.

// В поле Areq.threeDSCompInd Сервис указывает значение Y. Это признак успешного

// завершения 3DS-метода

// В поле Areq.notificationUrl Сервис указывает адрес возврата

// [**https://testmerchant.ru/iframe.php**](https://testmerchant.ru/iframe.php)

// Сервис получает в ответ Ares со статусом C

// Сервис сообщает Web-приложению о начале явной аутентификации

HTTP/1.1 200 OK

Content-Type: application/json

{

“state”: “iframe”,  
 “iframeWidth”: 390,  
 “iframeHeight”: 400,  
 “url”: “**test-acs.com\3ds-2**”,  
 “post”:{  
 “creq”: “ABCD123…DEF234==”  
 }

}

// Web-приложение создаёт фрейм и начинается взаимодействие клиента и ACS

// После завершения этого взаимодействия ACS отправляет Сервису сообщение Rreq с

// результатом аутентификации

// Сервис отвечает сообщением Rres

// ACS перенаправляет клиента на адрес **https://testmerchant.ru/iframe.php**.

// В запросе ACS указывает HTTP-параметр cres.

// Web-приложение получает запрос на свой адрес возврата.

// В запросе есть HTTP-параметр cres. При необходимости Web-приложение может

// его декодировать и узнать общий результат аутентификации.

// Web-приложение показывает клиенту какую-то страницу и одновременно с этим

// отправляет запрос в Сервис, чтобы сообщить Сервису о возврате клиента

// в Web-приложение.

POST /api/v4/TESTPORTAL123/payment/1234567890ABCDEF/merchant-return HTTP/1.1

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

cres=3453543874874KNG1CbhBsjtVYuo6e9cYI3543547I7tdi90OS93LgX25lru=

// Это сообщение о возврате Клиента не несёт никакой полезной информации,

// потому что к моменту его получения Сервис уже получил и обработал сообщение Rreq

// Сервис знает, чем закончилась аутентификация, знает eci и cavv

// И даже не ждёт запроса на адрес /merchant-return, а выполняет авторизацию

// Поэтому Web-приложение может вместо отправки запроса /merchant-return отправить

// обычный запрос статуса.

// Однако для унификации обработки состояния iframe рекомендуется всегда при

// возврате Клиента после iframe-перенаправления выполнять запрос /merchant-return

// Сервис возвращает статус «Платёж в обработке»

HTTP/1.1 200 OK

Content-Type: application/json

{

«state»: «in\_progress»

}

// Через некоторое время Web-приложение снова запрашивает текущий статус транзакции

GET /api/v4/TESTPORTAL123/payment/1234567890ABCDEF HTTP/1.1

// Сервис возвращает финальный статус

HTTP/1.1 200 OK

Content-Type: application/json

{

“token”: “1234567890ABCDEF”,

“startedAt” : “1456407089068”,

“finishedAt” : “1456407089406”,

“state”: “result”,

“description”: “Sample”,

“merchant”: {

“name”: “Test Merchant”,

“logoUrl”: “https://testmerchant.ru/logo.png”,

“url”: <https://testmerchant.ru>,

“emailReceiptEnabled”: true

},

“src”: {

“type”: “card”,

“pan”: “500000xxxxxx0000”,

“paymentSystem”: “MASTERCARD”

},

“amount”: “123456”,

“currency”: “RUB”,

“result”: {

“status”: “SUCCESS”,

“extendedCode”: “OK”,

“trxId”: “1234567890”,

“responseCode”: “00”,

“rrn”: “501617048902”,

“approvalCode”: “437762”,

“orderStatus”: “REGISTERED”,  
 “orderStatusChangedAt”: “1456407089068”

},

“portalType”: “MOBILE”,

“type”: “ACQUIRING”

}

## Выполнение рекуррентного платежа

// Запрос магазином нового токена

POST /api/v4/TESTPORTAL123/token HTTP/1.1

// Ответ с токеном

HTTP/1.1 200 OK

Content-Type: application/json

{«token»:»1234567890ABCDEF»}

// Запрос магазином старта платежа

POST /api/v4/TESTPORTAL123/payment/1234567890ABCDEF/start HTTP/1.1

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

merchantId=test\_merchant

merchantTrx=trx-12345

accountId=test\_account

src.type=card\_id  
src.cardId=ABCDEF1234567890  
params.order\_id=123  
amount=123456

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

returnUrl=https://site.ru/c2c?id=1234567890ABCDEF

// Сервис возвращает статус

HTTP/1.1 200 OK

Content-Type: application/json

{

“token”: “1234567890ABCDEF”,

“startedAt” : “1456407089068”,

“finishedAt” : “1456407089406”,

“state”: “result”,

“params”: {

“order\_id”: “123”

}

“merchant”: {

“name”: “Test Merchant”,

“logoUrl”: “https://testmerchant.ru/logo.png”,

“url”: https://testmerchant.ru

},

“src”: {

“type”: “card”,

“pan”: “500000xxxxxx0000”,

“cardId”: “ABCDEF1234567890”,

“title”: “card\_123”,

“paymentSystem”: “MASTERCARD”

},

“amount”: “123456”,

“currency”: “RUB”,

“result”: {

“status”: “SUCCESS”,

“extendedCode”: “OK”,

“trxId”: “1234567890”,

“responseCode”: “00”,

“rrn”: “501617048902”,

“approvalCode”: “437762”,

“authorizationTime”: “0110122234”

},

“portalType”: “web”,

“type”: “ACQUIRING”

}

## Аутентификация держателя карты

// Запрос клиентским приложением нового токена

POST /api/v4/TESTPORTAL123/token HTTP/1.1

// Ответ с токеном

HTTP/1.1 200 OK

Content-Type: application/json

{«token»:»1234567890ABCDEF»}

// Запрос клиентским приложением старта верификации карты

POST /api/v4/TESTPORTAL123/card/verification/1234567890ABCDEF/start HTTP/1.1

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

merchantId=test\_merchant

merchantTrx=trx-12345

accountId=test\_account

type=card

pan=5000000000000000

expiry=1030

amount=123456

currency=RUB

description=Sample

returnUrl=https://site.ru/c2c?id=1234567890ABCDEF

state.redirect=post\_params

// Сервис предлагает выполнить переход для проведения 3DS

HTTP/1.1 200 OK

Content-Type: application/json

{

“state”: “redirect”,

“url”: “acs.issuer.com/3ds”,

“post”: {

“PaReq”: “IqjKNG1CbhBsjtVYuo6e9cYI7I7tdi90OS93LgX25lrxU=”,

“MD”: “87654321”,

“TermUrl”: “https://service.ru/api/v4/TESTPORTAL123/card/verification/1234567890ABCDEF/return”

}

}

// Браузер клиента переходит по указанному адресу

// Клиент проходит 3DS

// Результат этого прохождения возвращается Сервису

// Сервис «возвращает» клиента клиентскому приложению

HTTP/1.1 302 Moved Temporarily

Location: <https://site.ru/c2c?id=1234567890ABCDEF>

// Клиентское приложение запрашивает текущий статус транзакции

GET /api/v4/TESTPORTAL123/card/verification/1234567890ABCDEF HTTP/1.1

// Сервис возвращает статус

HTTP/1.1 200 OK

Content-Type: application/json

{

“token”: “1234567890ABCDEF”,

“startedAt”: “1456407089068”,

“finishedAt”: “1456407089406”,

“state”: “result”,

“result”: {

“status”: “SUCCESS”,

“extendedCode”: “OK”,

“trxId”: “1234567890ABCDEF”,

“3ds”: {

“cavv”: “123456123456123456”,

“xid”: “123456712345671234567”,

“eci”: “05”,

“status”:” “PARES\_Y”

}

}

}

## Регистрация карты

// Запрос клиентским приложением нового токена

POST /api/v4/TESTPORTAL123/token HTTP/1.1

// Ответ с токеном

HTTP/1.1 200 OK

Content-Type: application/json

{«token»: «1234567890ABCDEF»}

// Запрос клиентским приложением старта регистрации карты

POST /api/v4/TESTPORTAL123/card/registration/1234567890ABCDEF/start HTTP/1.1

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

merchantId=test\_merchant

merchantTrx=trx-12345

accountId=test\_account

src.type=card\_page

returnUrl=https://site.ru/c2c?id=1234567890ABCDEF

state.redirect=post\_params

// Сервис предлагает выполнить переход на страницу ввода параметров карты

HTTP/1.1 200 OK

Content-Type: application/json

{

“state”: “redirect”,

“url”: “http://test-card-page.html?pageType=reg&cardholder=optional&submitUrl=url1&declineUrl=url2”

}

// Браузер клиента переходит по указанному адресу

// Клиент вводит параметры карты

// Введённые данные передаются Сервису

// Сервис предлагает выполнить переход для проведения 3DS

HTTP/1.1 200 OK

Content-Type: application/json

{

“state”: “redirect”,

“url”: “acs.issuer.com/3ds”,

“post”: {

“PaReq”: “IqjKNG1CbhBsjtVYuo6e9cYI7I7tdi90OS93LgX25lrxU=”,

“MD”: “87654321”,

“TermUrl”: “https://service.ru/api/v4/TESTPORTAL123/card/registration/1234567890ABCDEF/return”

}

}

// Браузер клиента переходит по указанному адресу

// Клиент проходит 3DS

// Результат этого прохождения возвращается Сервису

// Сервис «возвращает» клиента клиентскому приложению

HTTP/1.1 302 Moved Temporarily

Location: <https://site.ru/c2c?id=1234567890ABCDEF>

// Клиентское приложение запрашивает текущий статус транзакции

GET /api/v4/TESTPORTAL123/card/registration/1234567890ABCDEF HTTP/1.1

// Сервис возвращает статус

HTTP/1.1 200 OK

Content-Type: application/json

{

“token”: “1234567890ABCDEF”,

“startedAt”: “1456407089068”,

“finishedAt”: “1456407089406”,

“state”: “result”,

“result”: {

“status”: “SUCCESS”,

“extendedCode”: “OK”,

“trxId”: “1234567890ABCDEF”,

“responseCode: “000”,  
 “orderStatus”: “registered”,  
 “cardId”: “123456789012”,

“paymentSystem”: “MASTERCARD”

}

}

# Приложение 1. Коды ошибок выполнения запросов

Таблица Коды ошибок выполнения запросов

| HTTP-код | Наименование ошибки | Описание |
| --- | --- | --- |
| 400  Bad Request | ACCESS\_DENIED | Отказано в доступе. Необходимо проверить правильность введенных данных или обратиться в службу технической поддержки. Параметры:   * **subject** – сущность, к которой не удалось получить доступ (PARTNER, PORTAL, MERCHANT, USER). * **reason** – причина отказа в доступе (VERSION, OPERATION, INVALID, PAUSED, EXPIRED, BLOCKED). * **failedAttempts** – количество неудачных попыток предъявления пароля/PIN. * **delay** – количество секунд до снятия сущности с паузы.   Пример:  [subject=USER, reason=BLOCKED] – профиль клиента заблокирован;  [subject=USER, reason=INVALID] – неверный Id клиента, Id сессии или PIN устройства. Попытка продолжить операцию в анонимном режиме, но начатаю в не анонимном режиме. Для ситуации с неверным PIN устройства в поле failedAttempts помещается значение счетчика попыток предъявления PIN.  [subject=USER, reason=PAUSED] – профиль поставлен на паузу для запрошенной операции.  [subject=USER, reason=OPERATION] – профилю клиента запрещено выполнение запрошенной операции;  [subject=PORTAL, reason=INVALID] – неверный Id портала или попытка использовать сессию, открытую для другого партнера;  [subject=PORTAL, reason=BLOCKED] – портал заблокирован;  [subject=PORTAL, reason=OPERATION] – запрошенная операция не настроена или к ней запрещен доступ с указанного портала;  [subject=PORTAL, reason=VERSION] – портал недоступен для указанной версии приложения или портал требует обязательного указания версии приложения (если в запросе версия приложения не указана);  [subject=PARTNER, reason=BLOCKED] – партнер заблокирован.  Cочетание [subject=USER, reason=EXPIRED] преобразуется в отдельный код ошибки SESSION\_EXPIRED. |
| SESSION\_EXPIRED | Время сессии истекло.  Пример:  {«error»:»SESSION\_EXPIRED»} |
| ALREADY\_REGISTERED | Номер телефона, e-mail или карта уже зарегистрированы в системе. Необходимо использовать уникальные данные для регистрации в системе. Параметр:   * **entity** – тип сущности (EMAIL, MSISDN, PROFILE, CARD)   Пример:  {“error”:”ALREADY\_REGISTERED”, “entity”:”EMAIL”} |
| CARD\_NOT\_SUPPORTED | Карта не поддерживается. Необходимо использовать установленный тип платежных инструментов.  Пример:  {“error”:”CARD\_NOT\_SUPPORTED”} |
| CARD\_RANGE\_BLOCKED | Диапазон карт заблокирован. Параметр:   * **side** – указывает, является ли карта источником или приёмником денежных средств (DST, SRC)   Пример:  {“error”:”CARD\_RANGE\_BLOCKED”, “side”: “SRC”} |
| CARD\_BLOCKED | Карта заблокирована. Параметр:   * **side** – указывает, является ли карта источником или приёмником денежных средств (DST, SRC)   Пример:  {“error”:”CARD\_BLOCKED”, “side”: “SRC”} |
| CARD\_EXPIRED | Истек срок действия карты. Параметр:   * **side** – указывает, является ли карта источником или приёмником денежных средств (DST, SRC)   Пример:  {“error”:”CARD\_EXPIRED”, “side”: “SRC”} |
| CARD\_NOT\_VERIFIED | Карта не верифицирована.  Пример:  {«error»:»CARD\_NOT\_VERIFIED»} |
| DUPLICATE\_TITLE | Название карты или шаблона не уникально. Необходимо использовать уникальное название карты и шаблона.  Пример:  {“error”:”DUPLICATE\_TITLE”} |
|
| EXTRA\_PARAMETR | В запросе указан лишний параметр. Необходимо проверить правильность запроса.  Пример:  {“error”:”EXTRA\_PARAMETER”,”param”:”otp”}  где param – имя параметра. |
| INVALID\_PARAMETER | Недопустимое значение параметра. Необходимо проверить правильность введенных параметров.  Пример:  {“error”:”INVALID\_PARAMETER”,  “param”:”phonenumber”, “detail”:”Wrong  phone number format!”, “ maxValue”:”maxValue”}  где param – имя параметра.  maxValue – параметр, который был превышен в запросе.  Также ошибка может означать проблемы с параметрами платежа при работе с шаблонами платежей. В этом случае параметра param в сообщении об ошибке отсутствует. |
| INVALID\_PASSWORD | Логин не существует или пароль неверен, или нужна двухфакторная авторизация. Необходимо проверить правильность введенных данных или обратиться в службу технической поддержки. Параметры:   * **failedAttempts** – количество неудачных попыток предъявления пароля; * **attemptsLeft** – количество оставшихся попыток. Указывается для операций, выполняемых в рамках сессии (смена/проверка пароля, удаление профиля). * **delay** – количество секунд до снятия пароля с паузы. Указывается, если пароль был поставлен на паузу после неудачной попытки предъявления пароля.   Пример:  {“error”:”INVALID\_PASSWORD”,”failedAttempts”:2} |
| INVALID\_PAYMENT\_STATE | Платеж имеет неподходящее для выполнения запрошенной операции состояние.  Возможные варианты: код используется, если отправка уведомления запрошена для незавершенной транзакции.  Пример:  {“error”:”INVALID\_PAYMENT\_STATE”} |
| INVALID\_OTP | Одноразовый пароль, отправленный на телефон/почту, не прошел проверку, или Сервис не отправлял клиенту одноразовый пароль. Необходимо запросить новый одноразовый пароль или обратиться в службу технической поддержки.  Пример:  {“error”:”INVALID\_OTP”,  “failedAttempts”:2,  «attemptsLeft»:1}  где failedAttempts – количество неудачных попыток предъявления OTP;  attemptsLeft – количество оставшихся попыток. |
| INVALID\_REQUEST | Запрос не прошел валидацию. Необходимо проверить правильность и наличие всех обязательных параметров в запросе.  Пример:  {“error”: “INVALID\_REQUEST”, “detail”: “May not be empty: ‘otp’”}  где detail – описание ошибки. |
| MISSING\_PARAMETER | В запросе отсутствует обязательный параметр. Необходимо проверить наличие всех обязательных параметров в запросе.  Пример:  {“error”:”MISSING\_PARAMETER”,”param”:”otp”}  где param – имя параметра. |
| OTP\_EXPIRED | Истек срок действия одноразового пароля. Необходимо запросить новый одноразовый пароль.  Пример:  {«error»:»OTP\_EXPIRED»} |
| OTP\_PAUSED | Генерация нового одноразового пароля поставлена на паузу. После тайм-аута пауза будет снята автоматически.  Пример:  {“error”:”OTP\_PAUSED”, “delay”:135}  где delay – количество секунд до снятия с паузы. |
| OTP\_BLOCKED | Одноразовый пароль заблокирован из-за превышения количества попыток ввода. Необходимо запросить новый одноразовый пароль.  Пример:  {«error»:»OTP\_BLOCKED»} |
| PASSWORD\_EXPIRED | Срок действия пароля истек. Необходимо запросить новый одноразовый код.  Пример:  {“error”:”EXPIRED\_EXPIRED”} |
| PASSWORD\_PAUSED | Действие пароля приостановлено после неудачных попыток ввести правильный пароль. Необходимо проверить правильность введенных данных или обратиться в службу технической поддержки.  Параметры:   * **delay** – количество секунд до снятия с паузы.   Пример:  {“error”:”PASSWORD\_PAUSED”, “delay”:125} |
| PROFILE\_BLOCKED | Операция отклонена, т.к. профиль клиента заблокирован. Необходимо обратиться в службу технической поддержки.  Пример:  {«error»:»PROFILE\_BLOCKED»} |
| PROFILE\_NOT\_ACTIVATED | Профиль не активирован. Необходимо подтвердить регистрацию профиля с помощью одноразового пароля.  Пример:  {“error”:”PROFILE\_NOT\_ACTIVATED”} |
| TRANSACTION\_NOT\_SUCCESSFUL | Транзакция не успешна.  Пример:  {“error”:”TRANSACTION\_NOT\_SUCCESSFUL”} |
| UNSUPPORTED\_VERSION | Указанная версия протокола не поддерживается. Параметры:   * **minVersion** – минимальная поддерживаемая версия протокола. Праматер указывается, если в запросе указана недопустимо ранняя версия протокола; * **maxVersion** – максимальная поддерживаемая версия протокола. Праматер указывается, если в запросе указана недопустимо поздняя версия протокола.   Пример:  {“error”:”UNSUPPORTED\_VERSION”, “minVersion”:3} |
| UNEXPECTED\_REQUEST | Непредусмотренный запрос. После получения такой ошибки в ходе выполнения транзакции рекомендуется выполнить запрос статуса платежа (верификации/регистрации). Ответ сервера даст понимание того, что нужно делать для продолжения транзакции.  Запрос имеет корректный формат, но Сервис не может его обработать, так как это противоречит текущему состоянию транзакции. Код используется для сообщений, относящихся к выполнению платежа или регистрации карты.  Возможные варианты: пришел запрос на старт платежа с токеном, который уже используется для другого платежа. Пришел запрос подтверждения оферты, а платеж уже завершен.  Пример:  {«error»:»UNEXPECTED\_REQUEST»} |
| WEAK\_PASSWORD | Новый пароль не отвечает требованиям стойкости. Необходимо задать пароль в соответствии с установленными правилами.  Пример:  {«error»:»WEAK\_PASSWORD»} |
| PASSWORD\_ALREADY\_EXISTS | Новый пароль совпадает со старым (или старыми). Необходимо задать пароль в соответствии с установленными правилами.  Пример:  {«error»:» PASSWORD\_ALREADY\_EXISTS »} |
| MERCHANT\_BLOCKED | Магазин заблокирован.  Пример:  {“error”:”MERCHANT\_BLOCKED”} |
| TERMINAL\_NOT\_FOUND | Терминал с указанным идентификатором не найден.  Пример:  {“error”:”TERMINAL\_NOT\_FOUND”} |
| ACQUIRING\_TYPE\_NOT\_SUPPORTED\_BY\_MERCHANT | Данный тип эквайринга не поддерживается (или не настроен) для данного магазина.  Пример:  {“error”:”ACQUIRING\_TYPE\_NOT\_SUPPORTED\_BY\_MERCHANT”} |
| SRC\_CARD\_NOT\_SUPPORTED | Карта-источник не поддерживается или не подходит для данной операции. Код используется, если для карты-источника не удалось подобрать платежную схему, или бренд карты не поддерживается Сервисом.  Пример:  {“error”:”SRC\_CARD\_NOT\_SUPPORTED”} |
| CURRENCY\_NOT\_SUPPORTED | Неподдерживаемая валюта.  Пример:  {“error”:”CURRENCY\_NOT\_SUPPORTED”} |
| SRC\_TYPE\_NOT\_SUPPORTED | Тип источника средств не поддерживается.  Пример:  {“error”:”SRC\_TYPE\_NOT\_SUPPORTED”} |
| INVALID\_TRANSACTION\_STATE | Транзакция имеет неподходящее для выполнения запрошенной операции состояние. Код используется для сообщений, относящихся к выполнению платежа или регистрации карты.  Пример:  {“error”:”INVALID\_TRANSACTION\_STATE”} |
| TRANSACTION\_NOT\_FINISHED | Транзакция не завершена.  Пример:  {«error»:»TRANSACTION\_NOT\_FINISHED»} |
| CURRENCY\_MISMATCH | Валюта запроса не совпадает с ожидаемой.  Пример:  {«error»:»CURRENCY\_MISMATCH»} |
| AMOUNT\_TOO\_SMALL | Сумма, указанная в запросе, меньше допустимой.  Пример:  {“error”:”AMOUNT\_TOO\_SMALL”} |
| AMOUNT\_TOO\_BIG | Сумма, указанная в запросе, больше допустимой.  Пример:  {“error”:”AMOUNT\_TOO\_BIG”} |
| AMOUNT\_MISMATCH | Сумма, указанная в запросе, не совпадает с ожидаемой.  Пример:  {«error»:»AMOUNT\_MISMATCH»} |
| COMMISSION\_MISMATCH | Комиссия, указанная в запросе, не совпадает с ожидаемой.  Пример:  {“error”:” COMMISSION\_MISMATCH”} |
| TRANSACTION\_BUSY | Транзакция занята.  Данный код ошибки может быть получен в ответ на запрос возврата средств или завершения авторизации.  Он означает, что в данный момент для транзакции уже выполняется запрос и выполнение нового запроса невозможно. Клиентский интерфейс при получении данного кода должен подождать некоторое время и запросить актуальный статус транзакции.  Пример:  {«error»:» TRANSACTION\_BUSY «} |
| OPERATION\_NOT\_AVAILABLE | Операция недоступна.  Где reason – причина недоступности.  Пример:  {“error”:”OPERATION\_NOT\_AVAILABLE”, “reason”:”otp”} |
| 404 Not Found | CARD\_NOT\_FOUND | Карта не найдена. Необходимо зарегистрировать карту в системе. Параметр:   * **side** – указывает, является ли карта источником или приёмником денежных средств (DST, SRC)   Пример:  {“error”:”CARD\_NOT\_FOUND”} |
| EMAIL\_NOT\_FOUND | Е-mail не указан в профиле пользователя. Необходимо указать e-mail в профиле пользователя.  Пример:  {“error”:”EMAIL\_NOT\_FOUND”} |
| LOGIN\_NOT\_FOUND | Логин не найден. Необходимо зарегистрироваться в системе.  Пример:  {“error”:”LOGIN\_NOT\_FOUND”} |
| LOYALTY\_NOT\_FOUND | Сущность «лояльность» не найдена.  Пример:  {“error”:”LOYALTY\_NOT\_FOUND”} |
| MERCHANT\_NOT\_FOUND | ТСП не найдено. Необходимо обратиться в службу технической поддержки.  Пример:  {“error”:”Merchant not found”} |
| NOT\_FOUND | Сущность не найдена. Это может быть платежная транзакция, портал и т.п. сущности, для которых не заведены отдельные коды ошибок. Также ошибка используется, если в HTTP-запросе указан URL, который по формату не подходит для запросов Сервиса.  Пример:  {“error”:”NOT\_FOUND”} |
| PAYMENT\_NOT\_FOUND | Платеж с указанным алиасом не найден. Под платежом понимается описание платежа и платежной схемы. Необходимо проверить параметры запроса или обратиться в службу технической поддержки.  Пример:  {“error”:”PAYMENT\_NOT\_FOUND”} |
| TRANSACTION\_NOT\_FOUND | Транзакция не найдена. Подразумевается платежная транзакция или транзакция регистрации карты.  Пример:  {“error”:”TRANSACTION\_NOT\_FOUND”} |
| PORTAL\_NOT\_FOUND | Портал не найден.  Пример:  {“error”:”PORTAL\_NOT\_FOUND”} |
| 405 Method Not Allowed | METHOD\_NOT\_ALLOWED | HTTP-метод не поддерживается для указанного запроса. Необходимо использовать только допустимые HTTP-методы. В HTTP-ответе будет заголовок Allow, содержащий допустимые для запроса HTTP-методы.  Пример:  {“error”:”METHOD\_NOT\_ALLOWED”} |
| 500 Internal Server Error | INTERNAL\_ERROR | Внутренняя ошибка Сервиса. Необходимо обратиться в службу технической поддержки.  Пример:  {“error”:”INTERNAL\_ERROR”} |
|  |
|  |
|  |
| EXTERNAL\_SYSTEM\_ERROR | Ошибка при взаимодействии с внешней системой.  Пример:  {“error”:”EXTERNAL\_SYSTEM\_ERROR”} |
| 503 Service Unavailable | MAINTENANCE | Сервис временно недоступен. Проводятся регламентные работы.  Параметр:   * **scope** – область действия регламентных работ.   Возможные значения параметра:   * Server – регламентные работы на сервере. В этом случае сервер на любые запросы отвечает сообщением об ошибке с кодом MAINTENANCE и параметром scope=Server. Кроме того, в ответе сервера может содержаться следующее поле: finishTS – примерное время окончания работ (Unix time в ms). Параметр необязательный. * Payment – регламентные работы только в рамках одной платежной схемы. В данном случае сервер отвечает штатно на любые запросы, кроме запросов, связанных с проведением платежа по данной платежной схеме или комиссиями по платежу. На эти запросы он отвечает сообщением об ошибке с кодом MAINTENANCE и параметром scope=Payment.   Пример:  {“error”:”MAINTENANCE”, “scope”:”Payment” } |

# Приложение 2. Расширенные коды завершения платежа

Таблица Расширенные коды завершения платежа

| Наименование ошибки | Расширенный код результата платежа | Общий результат платежа | Описание |
| --- | --- | --- | --- |
| UNKNOWN | -1 | UNKNOWN | Результат платежа неизвестен (не определён). |
| OK | 0 | SUCCESS | Платеж завершен успешно. |
| SYSTEM\_ERROR | 2 | FAILED | Системная ошибка. Платеж завершился неуспешно из-за внутренней ошибки Сервиса. |
| COMMUNICATION\_ERROR | 3 | FAILED | Ошибка при взаимодействии с внешней системой. |
| BANK\_SYSTEM\_ERROR | 4 | FAILED | Системная ошибка на стороне банка |
| AMOUNT\_LIMIT\_EXCEEDED | 6 | FAILED | Превышен установленный лимит на общую сумму операций за отведенный период. |
| FREQUENCY\_LIMIT\_EXCEEDED | 7 | FAILED | Превышен установленный лимит на количество операций за отведенный период. |
| LIMIT\_EXCEEDED | 8 | FAILED | Превышен установленный лимит на количество или общую сумму операций за отведенный период. |
| CANCELED | 9 | DECLINED | Транзакция отменена инициатором платежа. |
| BLACK\_LIST | 10 | FAILED | Клиент находится в чёрном списке. |
| COMMISSION\_MISMATCH | 51 | FAILED | Расхождение в расчёте комиссии. |
| FM\_REJECTED | 104 | FAILED | Транзакция отклонена системой фрод-мониторинга (Сервиса или Банка). |
| FM\_REJECTED\_BY\_SCORE | 105 | FAILED | Оценка риска транзакции системой фрод-мониторинга превышает установленный порог. |
| BANK\_FM\_REJECTED | 106 | FAILED | Транзакция отклонена системой фрод-мониторинга Банка. |
| FM\_FAILED | 107 | FAILED | Ошибка при взаимодействии с системой фрод-мониторинга. |
| REFUND\_BY\_PARTNER | 200 | REFUND | Выполнен успешный возврат денежных средств по запросу партнера. |
| REFUND\_BY\_CUSTOMER | 201 | REFUND | Выполнен успешный возврат денежных средств по запросу клиента. |
| REFUND\_THROUGH\_CORRECTION | 202 | REFUND | В ходе сверки транзакция вручную помечена как "с возвратом денежных средств". |
| REFUND\_BY\_OPERATOR | 203 | REFUND | Выполнен успешный возврат денежных средств по запросу оператора административной консоли. |
| REFUND\_BY\_MERCHANT | 204 | REFUND | Выполнен успешный возврат денежных средств по запросу магазина. |
| REFUND\_BY\_SERVER | 205 | REFUND | Выполнен успешный возврат денежных средств по инициативе Сервиса. |
| PREAUTHORIZATION\_OK | 300 | INTERIM\_SUCCESS | Блокировка средств выполнена успешно. |
| FORMAT\_ERROR | 812 | FAILED | Неверный формат авторизационного сообщения. |
| REDIRECT\_REQUIRED | 1001 | FAILED | Невозможно продолжить транзакцию без  перенаправления (при state.redirect=no). |
| OFFER\_REQUIRED | 1003 | FAILED | Невозможно продолжить транзакцию без отображения оферты (при state.offer=never). |
| DECLINED\_BY\_CLIENT | 3014 | DECLINED | Транзакция отклонена клиентом. |
| INSUFFICIENT\_FUNDS | 3015 | FAILED | Недостаточно средств на платежной карте. |
| DECLINED\_BY\_ACQUIRER | 3016 | FAILED | Транзакция отклонена эквайером. |
| DECLINED\_BY\_ISSUER | 3017 | FAILED | Транзакция отклонена эмитентом. |
| CARD\_FAULT | 3018 | FAILED | Проблемы с картой, карточными данными |
| AUTHORIZATION\_FAILED | 3019 | FAILED | Ошибка авторизации. |
| CARD\_NOT\_SUPPORTED | 3020 | FAILED | Карта не поддерживается. |
| CARD\_DATA\_INVALID | 3021 | FAILED | Неверно указаны данные карты (например, CVV2/CVC2). |
| CARD\_NOT\_FOUND | 3022 | FAILED | Карта с указанным токеном не найдена |
| AUTHENTICATION\_TIMEOUT | 3023 | FAILED | Таймаут при выполнении аутентификации 3-D Secure. |
| AUTHENTICATION\_FAILED | 3024 | FAILED | Ошибка при выполнении аутентификации 3-D Secure. |
| AUTHORIZATION\_COMPLETION\_TIMEOUT | 3025 | FAILED | Истекло время завершения авторизации. |
| AUTHENTICATION \_REJECTED | 3026 | FAILED | Аутентификация отклонена клиентским устройством |
| AUTHENTICATION \_DECLINED | 3027 | DECLINED | Клиент отказался от аутентификации |
| STATE\_NOT\_REQUESTED | 3028 | FAILED | Не выполнен запрос статуса платежа или не выполнено перенаправление |
| CARD\_BLOCKED | 3100 | FAILED | Карта заблокирована. |
| CARD\_EXPIRED | 3101 | FAILED | Истек срок действия карты. |
| CARD\_LOST\_OR\_STOLEN | 3102 | FAILED | Карта утеряна или украдена. |
| CARD\_RANGE\_BLOCKED | 3103 | FAILED | Диапазон номеров карт для карты блокирован. |
| CARD\_LOST | 3104 | FAILED | Карта утеряна. |
| CARD\_STOLEN | 3105 | FAILED | Карта украдена. |
| CARD\_NOT\_EFFECTIVE | 3106 | FAILED | Карта недействительна. |
| CARD\_RESTRICTED | 3107 | FAILED | Карта ограничена. |
| CARD\_ TEMPORARY\_BLOCKED | 3110 | FAILED | Карта временно заблокирована. |
| OFFER\_TIMEOUT | 4010 | FAILED | Таймаут при подтверждении оферты. |
| OFFER\_DECLINED | 4024 | DECLINED | Оферта отклонена клиентом. |
| OFFER\_FAILED | 4010 | FAILED | Ошибка при подтверждении оферты. |
| CARD\_PAGE\_DECLINED | 4026 | DECLINED | Клиент отказался вводить данные платежной карты на CardPage. |
| CARD\_PAGE\_TIMEOUT | 4027 | FAILED | Таймаут при взаимодействии с клиентом на CardPage. |
| CARD\_PAGE\_FAILED | 4028 | FAILED | Ошибка при взаимодействии с клиентом на CardPage. |
| CPA\_REJECTED | 5001 | FAILED | Магазин ответил отказом на первой фазе  взаимодействия. |
| CPA\_FAILED | 5002 | FAILED | Ошибка при взаимодействии с магазином на первой фазе. |
| REGISTER\_PAYMENT\_FAILED | 5003 | FAILED | Ошибка при взаимодействии с магазином на завершающей фазе. |
| SUBSCRIPTION\_FAILED | 6101 | FAILED | Не удалось выполнить подписку на SMS-информирование по операциям с виртуальной картой. |
| DECLINED\_BY\_EXTERNAL\_SYSTEM | 9000 | FAILED | Транзакция отклонена внешней системой. |

# Приложение 3. Расширенные коды завершения верификации или регистрации карты

Таблица Расширенные коды завершения верификации карты

| Наименование ошибки | Расширенный код результата платежа | Общий результат платежа | Описание |
| --- | --- | --- | --- |
| UNKNOWN | -1 | UNKNOWN | Результат верификации неизвестен (не определен). |
| OK | 0 | SUCCESS | Верификация карты завершена успешно. |
| SYSTEM\_ERROR | -10 | FAILED | Системная ошибка. Верификация завершилась неуспешно из-за внутренней ошибки Сервиса. |
| CLIENT\_LOST | 2 | FAILED | Таймаут при взаимодействии с клиентом |
| CLIENT\_CANCEL | 3 | DECLINED | Клиент отказался от продолжения верификации карты |
| ATTEMPTS\_EXHAUSTED | 4 | FAILED | Исчерпаны попытки ввода контрольной суммы |
| AUTHENTICATION\_FAILED | 6 | FAILED | Ошибка при выполнении аутентификации Обычно этот код касается 3-D Secure-аутентификации |
| AUTHORIZATION\_FAILED | 7 | FAILED | Ошибка авторизации код ответа, полученный от банка, не является успешным |
| FM\_FAILED | 8 | FAILED | Ошибка при взаимодействии с системой фрод-мониторинга. |
| FM\_REJECTED | 9 | FAILED | Транзакция отклонена системой фрод-мониторинга (Сервиса или Банка) |
| FM\_REJECTED\_BY\_SCORE | 10 | FAILED | Оценка риска транзакции системой фрод-мониторинга превышает установленный порог. |
| LIMIT\_EXCEEDED | 12 | FAILED | Превышен установленный лимит на количество или общую сумму операций за период. Это может быть только лимит внешней системы, например, ПЦ банка |
| CARD\_REMOVED | 13 | FAILED | Во время транзакции карта была удалена |
| AUTHENTICATION \_REJECTED | 14 | FAILED | Аутентификация отклонена клиентским устройством |
| AUTHENTICATION \_DECLINED | 15 | DECLINED | Клиент отказался от аутентификации |
| REDIRECT\_REQUIRED | 1001 | FAILED | Невозможно продолжить транзакцию без  перенаправления (при state.redirect=no). |
| REGISTERED\_BY\_OTHER\_CLIENT | 1002 | FAILED | За время транзакции карта зарегистрирована другим клиентом |
| INSUFFICIENT\_FUNDS | 3015 | FAILED | Недостаточно средств на платежной карте. |
| CARD\_FAULT | 3018 | FAILED | Проблемы с картой, карточными данными |
| CARD\_DATA\_INVALID | 3021 | FAILED | Неверно указаны данные карты (например, CVV2/CVC2). |
| AUTHENTICATION\_TIMEOUT | 3023 | FAILED | Таймаут при выполнении аутентификации 3-D Secure. |
| STATE\_NOT\_REQUESTED | 3024 | FAILED | Не выполнен запрос статуса транзакции или не выполнено перенаправление |
| CARD\_BLOCKED | 3100 | FAILED | Карта заблокирована. |
| CARD\_EXPIRED | 3101 | FAILED | Истек срок действия карты. |
| CARD\_LOST\_OR\_STOLEN | 3102 | FAILED | Карта утеряна или украдена. |
| CARD\_PAGE\_DECLINED | 4026 | DECLINED | Клиент отказался вводить данные платежной карты на CardPage. |
| CARD\_PAGE\_TIMEOUT | 4027 | FAILED | Таймаут при взаимодействии с клиентом на CardPage. |
| CPA\_REJECTED | 5001 | FAILED | Магазин ответил отказом на первой фазе  взаимодействия. |
| CPA\_FAILED | 5002 | FAILED | Ошибка при взаимодействии с магазином на первой фазе. |

# Приложение 4. Статусы завершения аутентификации

Таблица Статусы завершения аутентификации 3DS v1

| Обозначение | Описание |
| --- | --- |
| VERES\_N | Карта не участвует в 3DS |
| VERES\_U | DS не удалось выполнить первый шаг аутентификации (VEReq/VERes) |
| PARES\_Y | Аутентификация выполнена успешно |
| PARES\_A | Аутентификация не выполнена, но попытка проведения аутентификации засчитана |
| PARES\_N | Аутентификация была неуспешной |
| PARES\_U | ACS не удалось выполнить второй шаг аутентификации (PAReq/PARes) |
| DS\_FAIL | Ошибка при взаимодействии с DS |
| DS\_ERROR | От DS получено сообщение об ошибке |
| ACS\_FAIL | Ошибка при взаимодействии с ACS |
| ACS\_ERROR | От ACS получено сообщение об ошибке |
| ACS\_TIMEOUT | Таймаут взаимодействия с ACS |
| BAD\_SIGNATURE | Подпись PARes не прошла валидацию |
| WITHOUT\_3DS | Транзакция выполнялась без аутентификации |

Таблица Статусы завершения аутентификации 3DS v2.1.0 и v2.2.0

| Обозначение | Статус ARes | Статус RReq | Описание |
| --- | --- | --- | --- |
| CHALLENGE\_MANDATED\_BY\_ACS | C | - | 3DS Server указал в AReq, что требуется провести неявную аутентификацию, или аутентификация выполняется для канала 3RI. |
| CHALLENGE\_MANDATED\_BY\_REQUESTOR | Y или A | - | 3DS Server указал в AReq, что требуется провести только явную аутентификацию |
| CHALLENGE\_IMPOSSIBLE\_BY\_REQUESTOR | С | - | ACS указал в ARes тип интерфейса, который не поддерживается в SDK |
| CHALLENGE\_CANCELLED\_BY\_RBA | C | - | Авторизация на основе оценки уровня риска |
| ARES\_FAILED | - | - | Произошла ошибка при отправке запроса AReq в DS, или при парсинге ответа от DS |
| ARES\_ERROR | - | - | В ответ на запрос AReq получено сообщение Error |
| ARES\_TIMEOUT | - | - | Истёк таймаут ожидания ответа на AReq |
| RREQ\_FAILED | C | - | Произошла ошибка при парсинге RReq |
| RREQ\_ERROR | C | - | Вместо RReq пришло сообщение Erro |
| RREQ\_TIMEOUT | C | - | Истёк таймаут ожидания RReq |
| ARES\_Y | Y | - | Успешная аутентификация |
| ARES\_N | N | - | Клиент не прошёл аутентификацию |
| ARES\_U | U | - | Аутентификацию провести не удалось |
| ARES\_A | A | - | Аутентификация не выполнена, но попытка проведения аутентификации засчитана |
| ARES\_R | R | - | Эмитент отказал в проведении аутентификации |
| RREQ\_Y | C | Y | Успешная аутентификация |
| RREQ\_N | C | N | Клиент не прошёл аутентификацию |
| RREQ\_U | C | U | Аутентификацию провести не удалось |
| RREQ\_A | C | A | Аутентификация не выполнена, но попытка проведения аутентификации засчитана |
| RREQ\_R | C | R | Эмитент отказал в проведении аутентификации |
| RREQ\_N\_CANCEL | C | N | Клиент прервал выполнение аутентификации |
| RREQ\_U\_CANCEL | C | U | Клиент прервал выполнение аутентификации |
| RREQ\_R\_CANCEL | C | R | Клиент прервал выполнение аутентификации |

Таблица Дополнительные статусы завершения аутентификации 3DS v2.2.0

| Обозначение | Статус ARes | Статус RReq | Описание |
| --- | --- | --- | --- |
| RREQ\_FAILED\_WITH\_ARES\_D | D | - | Произошла ошибка при парсинге RReq |
| RREQ\_ERROR\_WITH\_ARES\_D | D | - | Вместо RReq пришло сообщение Erro |
| RREQ\_TIMEOUT\_WITH\_ARES\_D | D | - | Истёк таймаут ожидания RReq |
| RREQ\_Y\_WITH\_ARES\_D | D | Y | Успешная аутентификация |
| RREQ\_N\_WITH\_ARES\_D | D | N | Клиент не прошёл аутентификацию |
| RREQ\_U\_WITH\_ARES\_D | D | U | Аутентификацию провести не удалось |
| RREQ\_A\_WITH\_ARES\_D | D | A | Аутентификация не выполнена, но попытка проведения аутентификации засчитана |
| RREQ\_R\_WITH\_ARES\_D | D | R | Эмитент отказал в проведении аутентификации |
| RREQ\_N\_CANCEL\_WITH\_ARES\_D | D | N | Клиент прервал выполнение аутентификации |
| RREQ\_U\_CANCEL\_WITH\_ARES\_D | D | U | Клиент прервал выполнение аутентификации |
| RREQ\_R\_CANCEL\_WITH\_ARES\_D | D | R | Клиент прервал выполнение аутентификации |
| UNEXPECTED\_STATUS\_D | D | - | 3DS Server указал в AReq, что не будет использовать Decoupled-аутентификацию. |
| UNEXPECTED\_STATUS\_I | I | - | 3DS Server указал в AReq, что проводит обычную, а не информационную аутентификацию |

# Приложение 5. Пример Open API запроса на старт In-App операции

Соответствие полей запроса Open API и данных In-App Payload:

| **Поле Open Api** | **Поле In-App** | **Примечание** | **Пример значения** |
| --- | --- | --- | --- |
| merchantId | - | merchantId, который зарегистрирован в PGA | merchant123456 |
| merchantTrx | orderId | Уникальный номер заказа/транзакции на стороне ТСП, тот же самый, что использовался при открытии приложения Mir Pay | order123456789 |
| src.pan | tan | Token Account Number – токенизированный номер карты | 2200123456789123486 |
| src.expiry, вторые две цифры | tey | Token Expiration Year – Год срока действия токенизированной карты | 29 |
| src.expiry, первые две цифры, дополняется нулем, если месяц состоит из одной цифры 3 -> 03 | tem | Token Expiration Month – месяц срока действия токенизированной карты | 3 |
| src.cavv | cav | Authentication Value (NSPK-InApp криптограмма) | 3D6FC826A08C82B89780029F69670FDCF2998 |
| src.transId | transId | Уникальный идентификатор In-Application операции, сформированный НСПК | 5ab52487-177f-464b-b695-2954ffc4a13 |
| amount | sum | сумма платежа в минорных единицах валюты(копейках) | 1000 |
| currency | cur | Валюта платежа | 643 |

Пример OpenAPI-запроса:

POST /api/v4/<portal\_id>/payment/<token>/start HTTP/1.0

Content-Type: application/json

merchantId=merchant123456

description=Mir Pay decrypted token without CPA

merchantTrx=order123456789

src.type=mir\_pay

src.pan=2200123456789123486

src.expiry=0329

src.cavv=3D6FC826A08C82B89780029F69670FDCF2998

src.transId=5ab52487-177f-464b-b695-2954ffc4a13

amount=1000

currency=643

state.redirect=no

# Приложение 6. Соответствие кодов ПЦ и Open Api при завершении платежа

Таблица . Соответствие кодов ПЦ и Open Api при завершении платежа

| Расширенный код | Код ответа  ПЦ | Расширенный код результата платежа  Open Api | Общий результат платежа | Описание |
| --- | --- | --- | --- | --- |
| OK | 000 | 0 | SUCCESS | Успешная транзакция |
| ERROR | 005 | 2 | FAILED | Системная ошибка. Повторите платеж позднее |
|  | 110 | Системная ошибка. Повторите платеж позднее |
|  | 117 | Системная ошибка. Повторите платеж позднее |
|  | 126 | Системная ошибка. Повторите платеж позднее |
|  | 127 | Системная ошибка. Повторите платеж позднее |
|  | 128 | Системная ошибка. Повторите платеж позднее |
|  | 320 | Ошибка обработки CVV2. Обратитесь к банку-эмитенту |
|  | 871 | Ошибка обработки CVV2. Обратитесь к банку-эмитенту |
|  | 872 | Ошибка обработки CVV2. Обратитесь к банку-эмитенту |
|  | 902 | Ошибка проведения платежа. Обратитесь к банку-эмитенту |
|  | 903 | Системная ошибка. Повторите платеж позднее |
|  | 904 | Системная ошибка. Повторите платеж позднее |
|  | 907 | Банк-эмитент недоступен |
|  | 909 | Ошибка проведения платежа. Обратитесь к банку-эмитенту |
|  | 910 | Банк-эмитент недоступен |
|  | 911 | Системная ошибка. Повторите платеж позднее |
|  | 912 | Банк-эмитент недоступен |
|  | 913 | Ошибка проведения платежа. Обратитесь к банку-эмитенту |
|  | 914 | Системная ошибка. Повторите платеж позднее |
|  | 915 | Системная ошибка. Повторите платеж позднее |
|  | 920 | Системная ошибка. Повторите платеж позднее |
|  | 923 | Системная ошибка. Повторите платеж позднее |
|  | 005 | Системная ошибка. Повторите платеж позднее |
|  | 110 | Системная ошибка. Повторите платеж позднее |
|  | 117 | Системная ошибка. Повторите платеж позднее |
| INSUFFICIENT\_FUNDS | 116 | 3015 | FAILED | Недостаточно средств |
| DECLINED\_BY\_ISSUER | 103 | 3017 | FAILED | Обратитесь к банку-эмитенту |
|  | 105 | Обратитесь к банку-эмитенту |
|  | 107 | Обратитесь к банку-эмитенту |
|  | 109 | Запрос отклонен банком-эмитентом |
|  | 118 | Запрос отклонен банком-эмитентом |
|  | 119 | Запрос отклонен банком-эмитентом |
|  | 120 | Запрос отклонен банком-эмитентом |
|  | 121 | Запрос отклонен банком-эмитентом |
|  | 123 | Запрос отклонен банком-эмитентом |
|  | 203 | Обратитесь к банку-эмитенту |
|  | 204 | Операции по счёту ограничены. Обратитесь к банку-эмитенту |
| 205 | Обратитесь к банку-эмитенту |
| 905 | Обратитесь к банку-эмитенту |
| 908 | Запрос отклонен банком-эмитентом |
| CARD\_FAULT | 100 | 3018 | FAILED | Карта не обслуживается. Обратитесь к банку-эмитенту |
|  | 104 | Карта не обслуживается. Обратитесь к банку-эмитенту |
|  | 111 | Карта не обслуживается. Обратитесь к банку-эмитенту |
|  | 125 | Карта испорчена. Обратитесь к банку-эмитенту |
|  | 200 | Карта недействительна. Обратитесь к банку-эмитенту |
|  | 202 | Карта недействительна. Обратитесь к банку-эмитенту |
|  | 940 | Карта не обслуживается. Обратитесь к банку-эмитенту |
| AUTHORIZATION\_FAILED | 3019 | 3019 | FAILED | Авторизация неуспешная |
| CARD\_BLOCKED | 106 | 3100 | FAILED | Карта блокирована. Обратитесь к банку-эмитенту |
|  | 206 | Карта блокирована. Обратитесь к банку-эмитенту |
| CARD\_EXPIRED | 101 | 3101 | FAILED | Карта просрочена. Обратитесь к банку-эмитенту |
|  | 201 | Карта просрочена. Обратитесь к банку-эмитенту |
| CARD\_LOST\_OR\_STOLEN | 208 | 3102 | FAILED | Карта утеряна. Обратитесь к банку-эмитенту |
|  | 209 | Карта украдена. Обратитесь к банку-эмитенту |
| CARD\_LOST\_OR\_STOLEN | 208 | 3104 | FAILED | Карта утеряна. Обратитесь к банку-эмитенту |
| CARD\_LOST\_OR\_STOLEN | 209 | 3105 | FAILED | Карта украдена. Обратитесь к банку-эмитенту |
| CARD\_FAULT | 200 | 3106 | FAILED | Карта недействительна. Обратитесь к банку-эмитенту |