

Ipay Merchant API Manual

BPC IPay - 1.55.1.0

Оглавление

1	О документе	4
2	Терминология.....	4
3	Подключение.....	5
4	Рекомендации по безопасности и средствам разработки.....	6
5	Взаимодействие	6
5.1	Схема с вводом реквизитов на стороне Платежного Шлюза	7
5.1.1	С 3D secure аутентификацией.....	8
5.1.2	Без 3D secure аутентификации	10
5.2	Схема с вводом реквизитов на стороне Магазина.....	13
5.2.1	С 3D secure аутентификацией.....	13
5.2.2	Без 3D secure аутентификации	15
5.3	Реализация схем на стороне магазина.....	16
5.4	Жизненный цикл заказа.....	17
5.5	Состояния заказа.....	20
5.6	Правила переходов.....	20
5.6.1	Состояние CREATED.....	20
5.6.2	Состояние DECLINED.....	20
5.6.3	Состояние APPROVED	21
5.6.4	Состояние DEPOSITED.....	21
5.6.5	Состояние REVERSED	21
5.6.6	Состояние REFUND.....	22
5.7	История заказа.....	22
6	Алгоритм действий для подключения к платежному шлюзу	23
7	Реализация API через интерфейс REST	23
7.1	Запросы, используемые при одностадийной оплате.....	24
7.1.1	Запрос регистрации заказа	24
7.1.2	Запрос оплаты заказа.....	26
7.1.3	Запрос отмены оплаты заказа.....	30
7.1.4	Запрос возврата средств оплаты заказа	31
7.1.5	Запрос состояния заказа	32
7.1.6	Запрос проверки вовлечённости карты в 3DS	36
7.2	Запросы, используемые при двухстадийной оплате	37
7.2.1	Запрос регистрации заказа с предавторизацией.....	37
7.2.2	Запрос оплаты заказа.....	40
7.2.3	Запрос завершения оплаты заказа.....	42
7.2.4	Запрос отмены оплаты заказа.....	43
7.2.5	Запрос возврата средств оплаты заказа	45
7.2.6	Запрос состояния заказа	46
7.2.7	Расширенный запрос состояния заказа.....	47
7.2.8	Запрос проверки вовлечённости карты в 3DS	51
8	Оформление платежной страницы	53

8.1 Требования к коду страницы платёжного интерфейса.	53
8.2 Требования к платёжной странице.....	54
8.2.1 Название страницы.....	54
8.2.2 Заголовок страницы.....	54
8.2.3 Тело страницы	55
8.2.4 Платежная форма	55
8.3 Требования к странице ошибок	57
8.3.1 Название страницы.....	57
8.3.2 Заголовок страницы.....	57
8.3.3 Тело страницы	58
9 Координаты подключения	59
10 Тестовые карты.....	60
11 Описание функционала связок	60
11.1 Общее описание	60
11.2 Технические особенности	61
11.3 Запросы по связкам через интерфейс REST	61
11.3.1 Создание связки.....	61
11.3.2 Запрос проведения платежа по связкам	61
11.3.3 Запрос деактивации связки.....	62
11.3.4 Запрос активации связки	63
11.3.5 Запрос списка возможных связок для мерчанта	64
12 Рекуррентные платежи	66
13 P2P платежи.....	67
14 P2Pcredit платежи	68
15 Использование связок при P2P и P2P_credit	68
16 3DS2 with card on client side	69
17 3DS2 with Binding on client side	74
18 Приложения.....	76
18.1 Приложение №1. Список значений action code.....	76

1 О документе

Настоящий документ описывает принципы подключения и программные интерфейсы платежного шлюза.

2 Терминология

- Магазин – торгово-сервисное предприятие (ТСП), продающее товары или оказывающее услуги через интернет-сайт.
- Банковская Карта – карта платежных систем (ArCa, VISA, MasterCard и пр.).
- Банк-эмитент – банк, выпустивший банковскую карту клиента.
- МПС – Международная платежная система (например, Visa или MasterCard)
- шлюз Банка-эквайера (Платежный Шлюз) – автоматическая система, предоставляющая магазину принимать, а Клиенту отправлять платежи через Интернет с использованием банковских карт. Оператором Платежного шлюза является компания «Армениан Кард».
- Банк-эквайер – банк, который реализует и эксплуатирует платежный шлюз.
- 3D Secure – технология МПС VISA, позволяющая дополнительно авторизовать пользователя средствами банка-эмитента.
- SecureCode – технология МПС MasterCard, позволяющая дополнительно авторизовать пользователя средствами банка-эмитента. Технологически равносильна 3D Secure. В тексте ниже при упоминании 3D Secure будем иметь ввиду и SecureCode.
- Merchant Plugin Interface (MPI) – технологический компонент 3D Secure и SecureCode, который может быть размещен на стороне ПС или на стороне магазина
- ACS – Access Control Server, элемент инфраструктуры 3D Secure, обеспечивающий валидацию плательщика на стороне банка-эмитента.
- Платежная форма – HTML-страница, которая используется клиентом для ввода реквизитов платежа.
- Платежные реквизиты – реквизиты, используемые пользователем для оплаты заказа. Обычно, это имя и фамилия Плательщика, номер карты, срок действия, код CVC2.
- Плательщик – физическое лицо совершающее платеж по своей карте за услуги Мерчанта в интернет магазине Мерчанта.
- Связка – соответствие между Плательщиком и платежными реквизитами карты (номер карты, срок действия карты), которое может быть использовано для проведения автоматических платежей без дополнительных действий со стороны Плательщика
- Заказ – элементарная сущность системы, описывает заказ в некотором интернет-магазине или его аналоге. У любого заказа есть сумма.
- Одностадийный платеж – операция по оплате товаров/услуг, совершенная через Интернет с использованием банковских карт, которая не требует дополнительного подтверждения.
- Двухстадийный платеж – операция по оплате товаров/услуг, совершенная через Интернет с использованием банковских карт, требующая дополнительного подтверждения. Двухстадийный механизм работы позволяет разделить процесс проверки платежеспособности банковской карты (авторизация) и снятие денег (финансовое

подтверждение). На первой стадии двустадийного платежа происходит проверка платежеспособности банковской карты и блокирование средств на счету клиента.

- Отмена операции оплаты (Reversal) – снятие блокировки с денежных средств на карте покупателя. Данная функция доступна ограниченное время, точные сроки необходимо уточнять в банке.
- Возврат средств (Refund) – возврат денежных средств на карту покупателя в случае его отказа от получения товара (услуги) или его возврата. Операция возврата денежных средств выполняется после списания денежных средств со счета покупателя. Система позволяет проводить частичный возврат, а также возврат по частям. Общая сумма возвращенных средств не может быть больше чем сумма оригинального заказа.

3 Подключение

Для подключения к системе необходимо связаться с каким-либо из коммерческих банков, подключенных к системе «Армениан Кард». Несмотря на то, что оператором Платежного шлюза является компания «Армениан Кард», она не заключает договора с Магазинами и Плательщиками, все отношения выстраиваются через банков-участников.

Компания «Армениан Кард» предоставляет тестовую платформу, с помощью которой Магазин может проводить разработку и тестирование своей системы до того, как начать принимать реальные платежи. Для подключения к тестовой системе необходимо обратиться либо к банку-партнеру, либо к специалистам компании «Армениан Кард».

В результате подключения магазин получает:

- Логин – имя магазина в рамках платежного шлюза. Оно используется только для произведения операций посредством API.
- Пароль – пароль магазина в рамках платежного шлюза. Используется только для произведения операций посредством API
- Пару логин+пароль для доступа к личному кабинету

В случае использования связок (см. главу [Описание функционала связок](#)) магазин получает еще одну пару логин+пароль, которая будет использоваться только для проведения платежей по связкам посредством API.

В необязательном порядке Магазин предоставляет:

- HTML-страницу, с графикой и CSS и прочими подключаемыми объектами, которая показывает платежную форму. Требования к этой странице описаны в разделе [Оформление платежной страницы](#). Возможен вариант, когда банк-партнер магазина сам предоставляет платежную страницу для всех магазинов, с которыми он работает, это необходимо уточнять в банке-партнере.

4 Рекомендации по безопасности и средствам разработки

В процессе интеграции настоятельно рекомендуется использовать те средства разработки, методы и практики, которые соответствуют распространенным стандартам безопасности разработки ПО, например OWASP.

Рекомендуется использовать наиболее «свежие» версии программного обеспечения, средств разработки, максимально новые версии протоколов безопасности и пр.

Не рекомендуется использовать нестандартные подходы, номера портов и прочее, например использовать для [https](https://) номера портов отличные от 443.

Оператор Платежного Шлюза несет ответственность за безопасность технических и программных компонент используемых при эксплуатации Платежного шлюза, в связи с чем при возникновении угроз или уязвимостей, а также при устаревании технологий и протоколов, Оператор может в одностороннем порядке отключать поддержку уязвимых или устаревших технологий, что в свою очередь может привести к неработоспособности систем магазинов.

5 Взаимодействие

Работа платежного шлюза различается для случаев использования и неиспользования технологии аутентификации 3D secure.

По умолчанию все магазины подключенные к системе используют технологию 3D secure инициируют проведение платежей с использованием данной технологии. Однако, в зависимости от типа платежной карты, банка, выпустившего карту или по иным причинам, платежи могут завершаться без 3D secure аутентификации.

Система поддерживает 2 метода ввода платежных реквизитов:

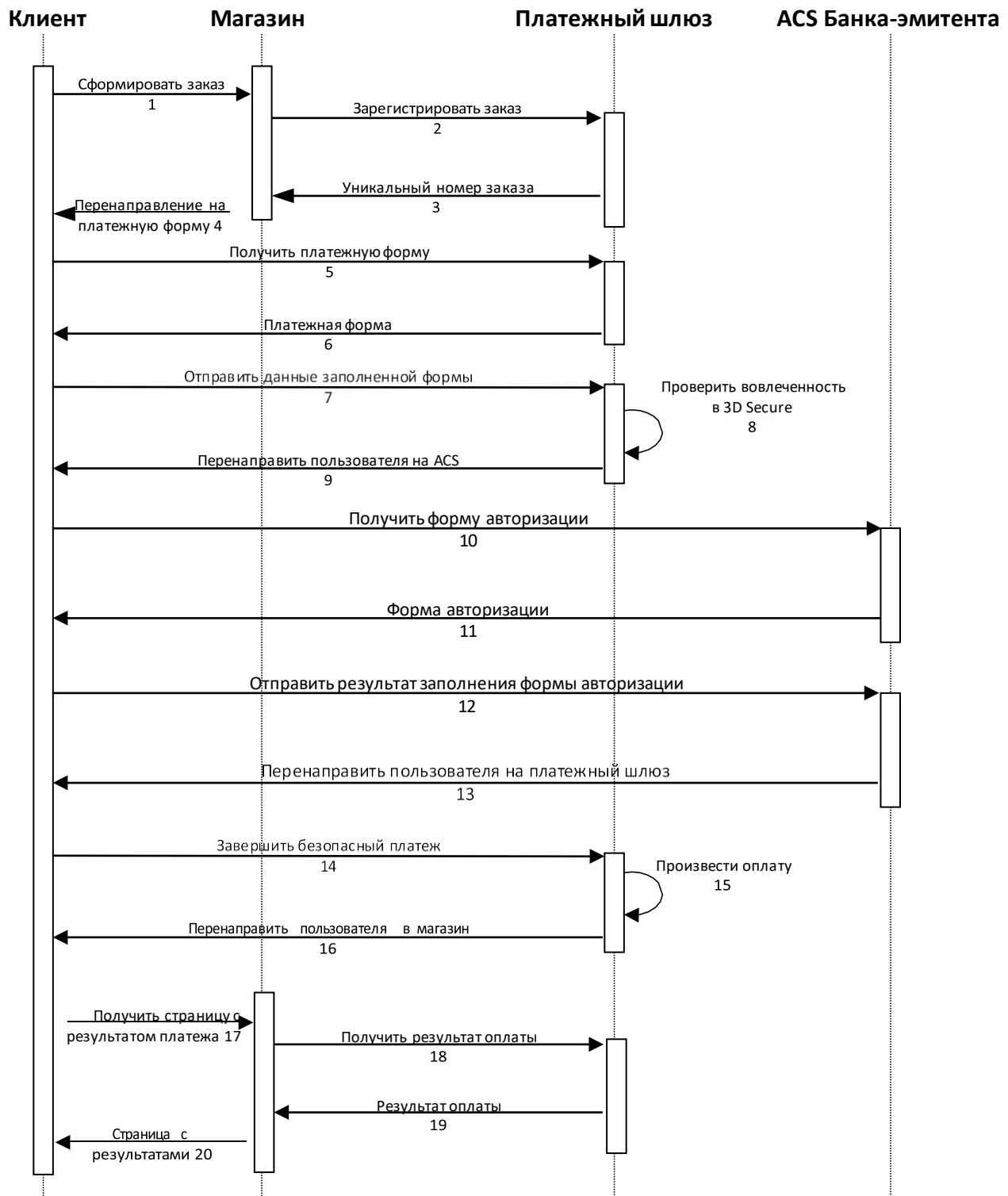
- **Данные вводятся на стороне Платежного шлюза.** При использовании данного метода Пользователь, в процессе проведения платежа перенаправляется Магазином на адрес Платежного шлюза. Платежные реквизиты вводятся на странице Платежного шлюза и остаются скрытыми для Магазина. Данный метод является самым распространенным.
- **Данные вводятся на стороне Магазина.** При использовании данного метода Магазин организует ввод Платежных реквизитов на своей странице и затем отправляет их Платежному шлюзу. Данный метод технически сложнее, и кроме того подразумевает наличие у Магазина сертификата соответствия стандарту безопасности индустрии платежных карт PCI DSS. Использование данного метода не допускается без наличия указанного сертификата и требует согласования с банком-партнером.

Для каждого из указанных методов возможны 2 схемы проведения платежа: с использованием и без использования 3D secure аутентификации. Как уже было сказано выше по умолчанию все магазины инициируют платежи как 3D secure, но по независящим от Магазина причинам платеж может завершиться как SSL (без использования 3D secure аутентификации). Сказанное не означает, что платеж проходит без

использования 3D secure по инициативе Магазина или по причине какой-либо ошибки со стороны Магазина. Для полноценного 3D secure необходимо, чтобы обе стороны: и Магазин, и Плательщик, были бы вовлечены в технологию 3D secure. Если сторона Плательщика не вовлечена – платеж может пройти без 3D secure аутентификации. В зависимости от банка-партнера Магазина и/или банка выпустившего карту Плательщика платеж, инициированный как 3D secure, но завершившийся без 3D secure аутентификации может быть отклонен. По данному вопросу настоятельно рекомендуется проконсультироваться с банком-партнером.

5.1 Схема с вводом реквизитов на стороне Платежного Шлюза

5.1.1 С 3D secure аутентификацией



Описание:

1. Клиент взаимодействует с магазином для создания заказа в магазине.
2. После подтверждения заказа клиентом, система магазина регистрирует заказ в платежном шлюзе. Для регистрации используются такие параметры как сумма списания, валюта списание, номер заказа в системе магазина, а также URL возврата клиента.
3. На запрос регистрации платежный шлюз возвращает уникальный идентификатор заказа в платежной системе и URL, на который надо перенаправить пользователя для получения платежной формы.
4. Система магазина передает браузеру клиента redirect на URL, полученный на шаге 3.
5. Браузер клиента открывает URL
6. В качестве страницы по указанному URL браузер клиента получает платежную форму
7. Пользователь заполняет полученную форму и отправляет данные на сервер платежного шлюза
8. Система проверяет вовлеченности карты в 3D Secure (SecureCode).
9. Шлюз отправляет браузеру клиента redirect на страницу ACS банка эмитента (этот шаг является необходимым для реализации 3DS)
10. Браузер клиента запрашивает у ACS форму авторизации пользователя (у каждого эмитента это делается по-своему)
11. ACS отправляет в браузер клиента эту форму, клиент ее видит
12. Клиент заполняет форму и отправляет ее в ACS
13. ACS обрабатывает заполненную форму и (в не зависимости от результата) передает браузеру redirect на URL страницы платежного шлюза. Вместе с этим URL передаются зашифрованные параметры результата авторизации
14. Браузер клиента запрашивает страницу платежного шлюза, передавая зашифрованные параметры результата авторизации
15. Платежный шлюз производит оплату (списание)
16. После проведения оплаты, платежный шлюз передает браузеру клиента URL возврата (указанный ранее при регистрации заказа магазином)
17. Браузер клиента запрашивает страницу с результатами оплаты у магазина.
18. Система магазина запрашивает платежный шлюз о статусе оплаты заказа (по внутреннему номеру в платежной системе)
19. Платежный шлюз возвращает статус оплаты
20. Система магазина передает в браузер клиента страницу с результатами оплаты.

В этой схеме шаги 18 и 19 не являются обязательными, магазин может не использовать их в работе.

5.1.2 Без 3D secure аутентификации

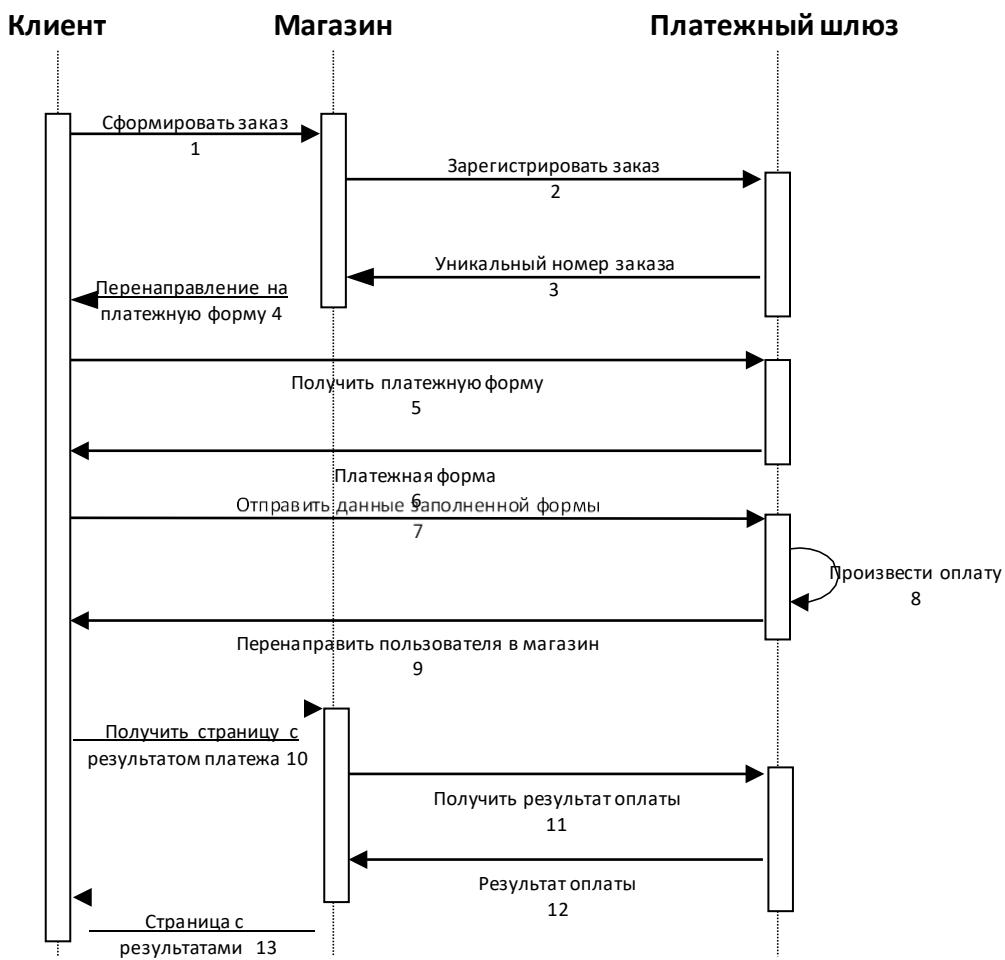


Схема №2

Описание:

- Клиент взаимодействует с магазином для создания заказа в магазине.
- После подтверждения заказа клиентом, система магазина регистрирует заказ в платежном шлюзе. Для регистрации используются такие параметры как сумма списания, валюта списание, номер заказа в системе магазина, а также URL возврата клиента.
- На запрос регистрации платежный шлюз возвращает уникальный идентификатор заказа в платежной системе и URL, на который надо перенаправить пользователя для получения платежной формы.
- Система магазина передает браузеру клиента redirect на URL, полученный на шаге 3.
- Браузер клиента открывает URL
- В качестве страницы по указанному URL браузер клиента получает платежную форму
- Пользователь заполняет полученную форму и отправляет данные на сервер платежного шлюза
- Получив платежные реквизиты (номер карты и т.п.) система производит списание со счета клиента
- После проведения оплаты, платежный шлюз передает браузеру клиента URL возврата

(указанный ранее при регистрации заказа магазином)

10. Браузер клиента запрашивает страницу с результатами оплаты у магазина.
11. Система магазина запрашивает платежный шлюз о статусе оплаты заказа (по внутреннему номеру в платежной системе)
12. Платежный шлюз возвращает статус оплаты
13. Система магазина передает в браузер клиента страницу с результатами оплаты

Если по истечении отведенных на оплату 20 минут клиент не вернулся с платежного шлюза на страницу результатов оплаты магазина (на URL возврата клиента), то оплата считается неудачной.

Изменение статуса оплаты заказа может быть выполнено по запросу магазина вручную сотрудниками Банка после проверки состояния транзакции в банковских системах. В этом случае после подтверждения изменения статуса заказа магазин может выполнить повторный запрос о статусе оплаты заказа (шаги 18-19).

По завершении п.20 заканчивается взаимодействие между магазином и платежным шлюзом в on-line режиме. Дальнейшие операции по завершению платежа (в случае двухстадийных платежей), отмене платежа и возврату денежных средств проводятся в off-line режиме.

5.2 Схема с вводом реквизитов на стороне Магазина

5.2.1 С 3D secure аутентификацией

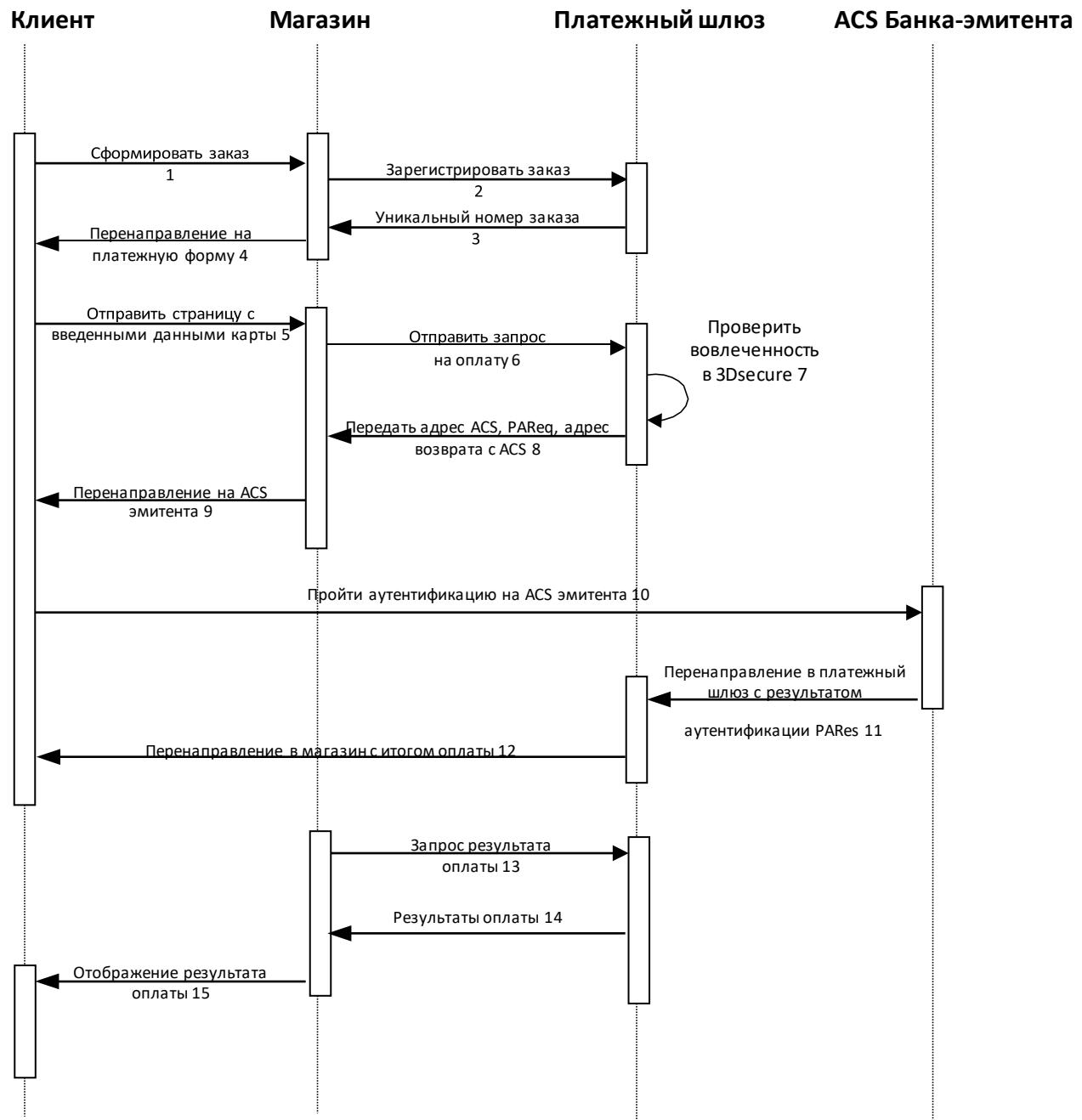


Схема №3.

Описание:

1. Держатель карты формирует заказ на сайте интернет-магазина и принимает решение оплатить товар банковской картой.

2. После подтверждения заказа клиентом, система магазина регистрирует заказ в платежном шлюзе. Для регистрации используются такие параметры как сумма списания, валютасписания, номер заказа в системе магазина, а также URL возврата клиента.
3. На запрос регистрации платежный шлюз возвращает уникальный идентификатор заказа в платежной системе и URL, на который надо перенаправить пользователя для получения платежной формы.
4. Интернет-магазин **игнорирует** URL присланный платежным шлюзом и переводит браузер держателя карты на **свою** платежную страницу для ввода данных по карте.
5. Держатель карты вводит данные своей банковской карты на платежной странице интернет-магазина и отправляет данные.
6. Магазин отправляет в платежный шлюз запрос на оплату заказа. Для оплаты используются такие параметры как номер заказа в платежном шлюзе (полученный на шаге 3), язык, а также реквизиты карты: PAN, CVC, год и месяц срока действия карты и имя владельца карты.
7. Получив платежные реквизиты (номер карты и т.п.) система проверяет вовлечение карты в 3d secure, в том числе с использованием Directory Server. **ВНИМАНИЕ!** В данном пункте рассматривается сценарий, когда карта вовлечена в 3d secure и платежный шлюз возвращает URL сервера ACS эмитента. До текущего шага магазин не может знать по какой схеме происходит платеж – с 3d secure аутентификацией или без неё. Именно на данном шаге магазин получает информацию, на основании которой определяется, какие шаги необходимо выполнять дальше. Наличие параметров acsUrl и paReq в ответе платежного шлюза означает, что необходимо выполнить 3d secure аутентификацию карты.

Шаги 8, 9, и 10 предполагают, что платеж проходит по сценарию, когда карта вовлечена в 3d secure.

8. Результат проверки отправляется в магазин (вовлеченность карты + URL сервера ACS эмитента, на котором клиенту необходимо пройти аутентификацию). Система магазина передает в браузер клиента перенаправление на ACS эмитента.
9. Держатель карты (клиент) проходит аутентификацию на ACS эмитента.
10. Проведя аутентификацию держателя карты, ACS эмитента перенаправляет клиента в платежный шлюз, передавая PARes. Платежный шлюз проводит авторизацию заказа при успешной аутентификации клиента на ACS, или отклоняет предавторизацию при неуспешной аутентификации клиента на ACS.
11. Клиент переходит на страницу магазина с итогами оплаты.
12. Магазин направляет в платежный шлюз запрос на статус оплаты заказа
13. Платежный шлюз возвращает результат авторизации заказа в магазин.
14. Магазин отображает клиенту страницу с результатами оплаты заказа.

5.2.2 Без 3D secure аутентификации

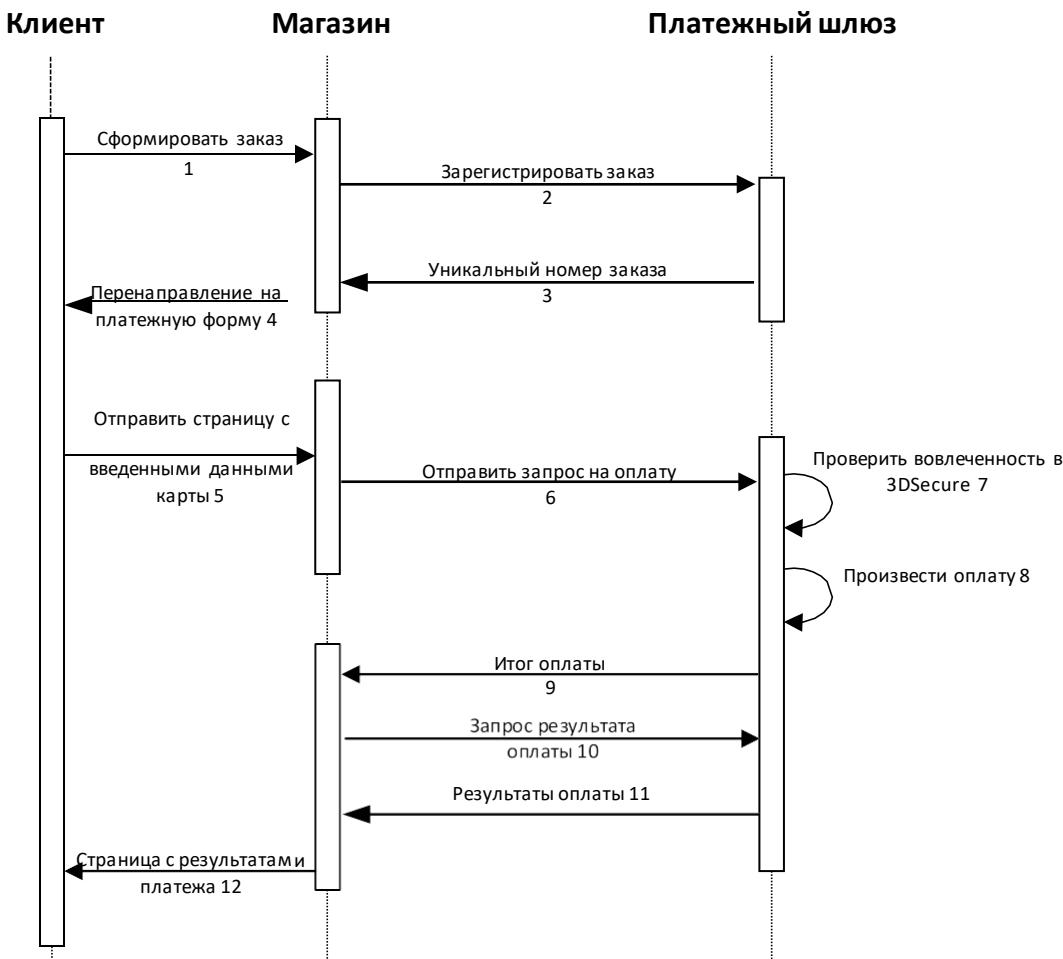


Схема №4

Описание:

1. Держатель карты формирует заказ на сайте интернет-магазина и принимает решение оплатить товар банковской картой.
2. После подтверждения заказа клиентом, система магазина регистрирует заказ в платежном шлюзе. Для регистрации используются такие параметры как сумма списания, валюта списания, номер заказа в системе магазина, а также URL возврата клиента.
3. На запрос регистрации платежный шлюз возвращает уникальный идентификатор заказа в платежной системе и URL, на который надо перенаправить пользователя для получения платежной формы.
4. Интернет-магазин переводит браузер держателя карты на свою платежную страницу для ввода данных по карте.
5. Держатель карты вводит данные своей банковской карты на платежной странице интернет-магазина и отправляет данные.

6. Магазин отправляет в платежный шлюз запрос на оплату заказа. Для оплаты используются такие параметры как номер заказа в платежном шлюзе (полученный на шаге 3), язык, а также реквизиты карты: PAN, CVC, год и месяц срока действия карты и имя владельца карты.
7. Получив платежные реквизиты (номер карты и т.п.) система проверяет вовлечение банка и карты в 3d secure. **ВНИМАНИЕ!** В данном пункте рассматривается сценарий, когда карта **НЕ** вовлечена в 3d secure и платежный шлюз **НЕ** возвращает URL сервера ACS эмитента. До текущего шага магазин не может знать по какой схеме происходит платеж – с 3d secure аутентификацией или без неё. Именно на данном шаге магазин получает информацию, на основании которой определяется, какие шаги необходимо выполнять дальше.
8. В случае невовлеченности банка или карты в 3d secure производит списание со счета клиента.
9. После проведения оплаты шлюз возвращает ответ об успешности или неуспешности оплаты. В ответе должны отсутствовать параметры acsUrl и paReq, так как карта не вовлечена в 3dsecure.
10. Система магазина запрашивает платежный шлюз о статусе оплаты заказа (по внутреннему номеру в платежной системе)
11. Платежный шлюз возвращает статус оплаты
12. Система магазина передает в браузер клиента страницу с результатами оплаты.

5.3 Реализация схем на стороне магазина

В рамках всех схем, между системами магазина и платежным шлюзом имеются автоматические взаимодействия:

- 2, 3 – на всех схемах,
- 18, 19 – на первой;
- 11 и 12 – на второй;
- 6 и 8 – на третьей;
- 13 и 14 – на третьей;
- 10 и 11 – на четвертой.

Эти взаимодействия синхронны, т.е. вызывающая сторона должна дождаться получения ответа или таймаута для продолжения работы, а также односторонни – всегда система магазина обращается к платежному шлюзу и никогда не наоборот. Чтобы разработчик системы магазина мог реализовать эти взаимодействия, система платежного шлюза представляет API из 2x запросов:

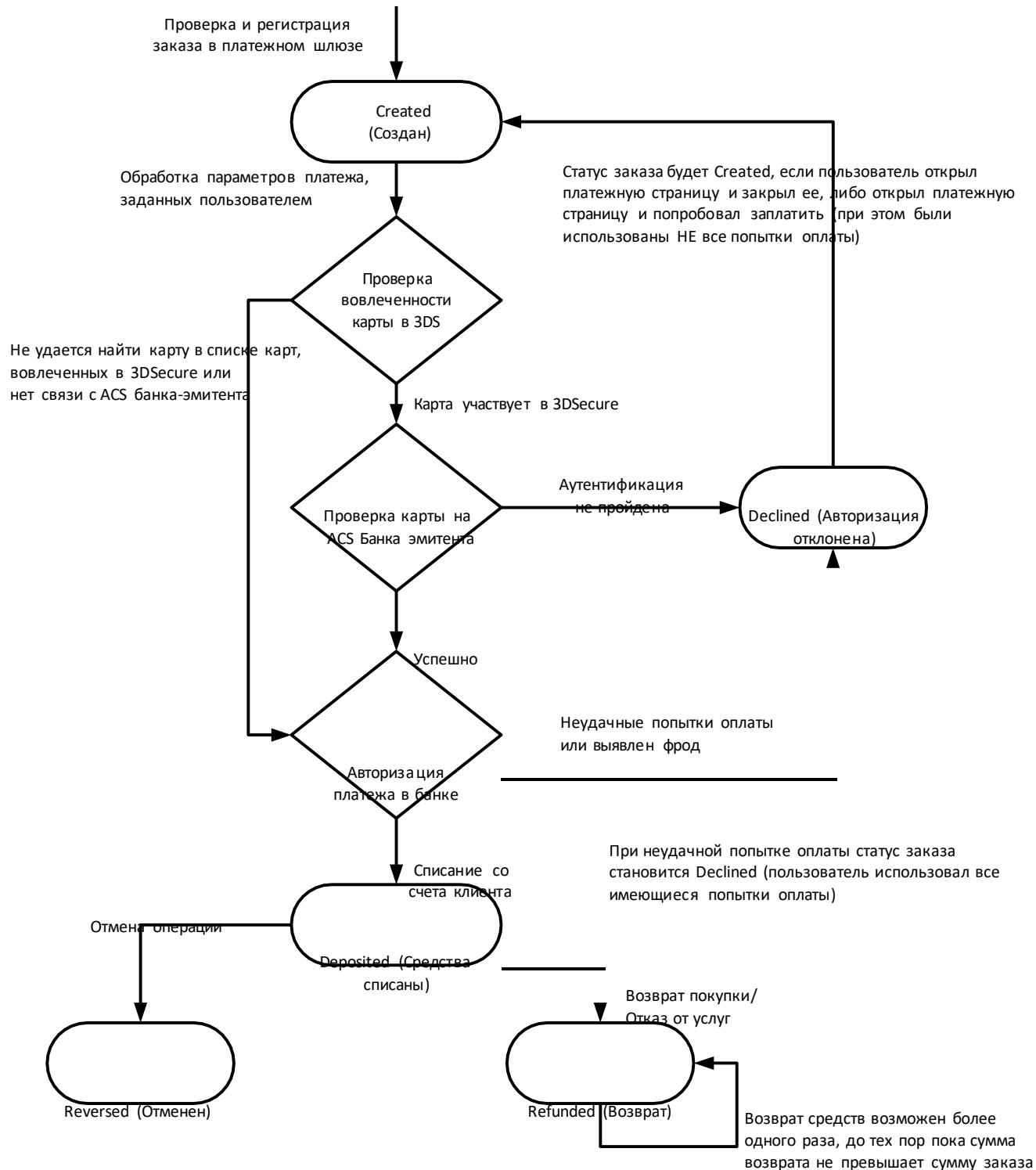
- Зарегистрировать заказ
- Получить информацию по состоянию заказа

Система предоставляет реализацию API через интерфейс REST.

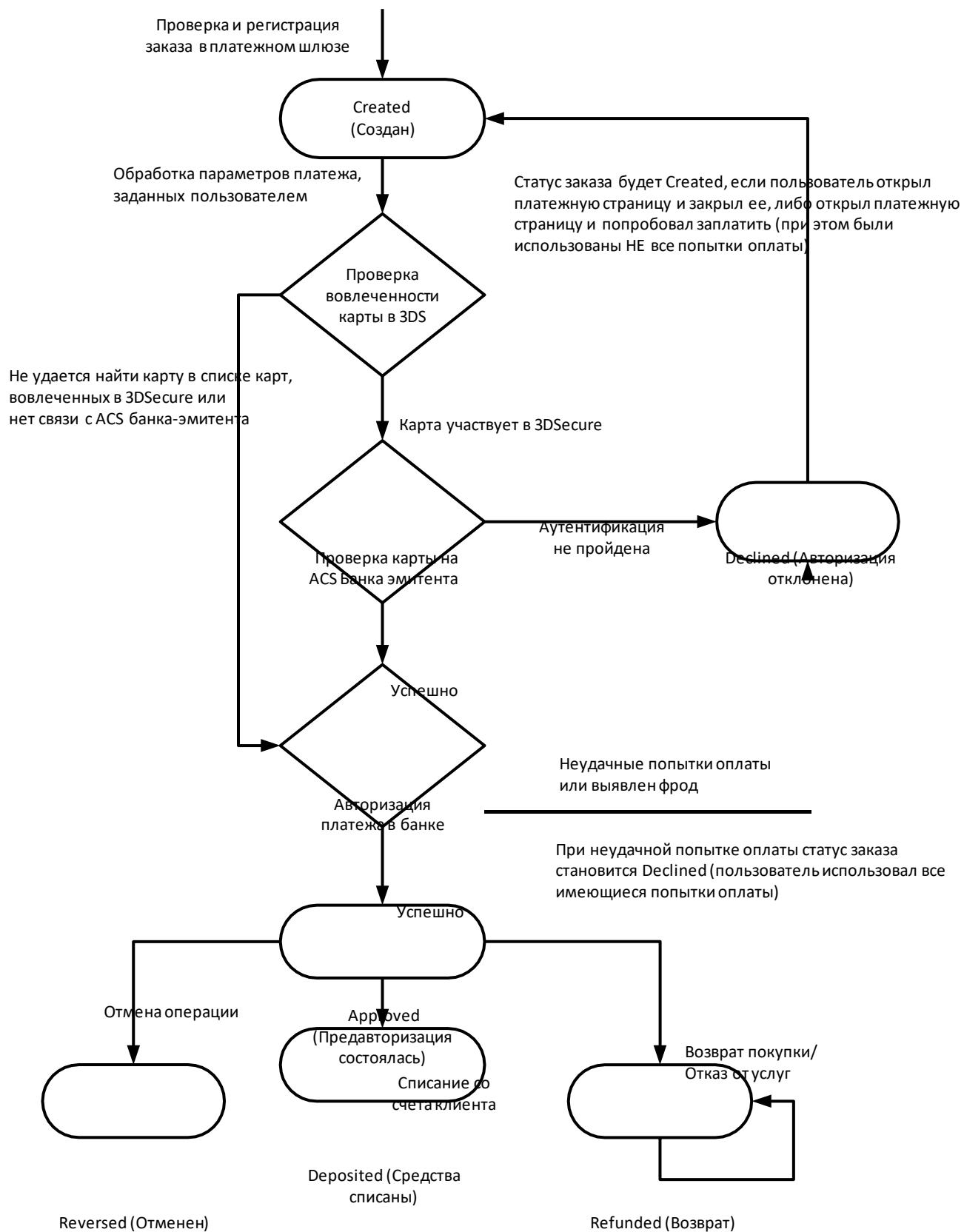
Примечание: С момента регистрации заказа у покупателя есть 20 минут для его оплаты. При попытке его оплаты по истечении этого срока будет выдаваться страница с ошибками.

5.4 Жизненный цикл заказа

Одностадийные платежи



Двухстадийные платежи



Возврат средств возможен более
одного раза, до тех пор пока сумма
возврата не превышает сумму заказа

Состояние заказа в системе характеризуется 2 параметрами:

- Состояние заказа
- Код ответа последний операции

Кроме того, все манипуляции с заказом отражаются в истории заказа.

5.5 Состояния заказа

Заказ может находиться в одном из следующих состояний:

Название в пользовательском интерфейсе	Внутреннее название	Описание
CREATED	started	Заказ создан
APPROVED	payment_approved	Пред-авторизация по заказу успешно состоялась
DECLINED	payment_declined	Пред-авторизация, либо авторизация по заказу отклонена
REVERSED	payment_void	Заказ отменен
DEPOSITED	payment_deposited	Средства по заказу списаны
REFUNDED	refunded	Средства по заказу возвращены

5.6 Правила переходов

5.6.1 Состояние CREATED

Состояние заказа будет Created, если пользователь:

- Открыл платежную страницу и закрыл ее,
- Открыл платежную страницу и попробовал заплатить, при этом были использованы НЕ все попытки оплаты (количество попыток оплаты задается в настройках системы), но не получилось.

Из состояния Created заказ может перейти в состояния Approved (в случае двухстадийного платежа), Deposited (в случае одностадийного платежа), либо Declined (в случае, если были использованы все попытки оплаты, установленных в настройках системы, и не удалось оплатить, либо если вес фрод = 100).

Примечание. Количество попыток оплаты задается администратором в консоли в разделе "Продавцы".

5.6.2 Состояние DECLINED

Состояние заказа будет Declined, если:

- Пользователь использовал все попытки (количество попыток задается в настройках системы) и попытки оказались неуспешными,
- Любая из попыток заплатить выявила фрод,

- Карта не прошла авторизацию на ACS банка-эмитента,
- Платёжная страница была открыта, попыток оплаты не было и время сессии истекло,
- Платёжная страница была открыта, использованы НЕ все попытки оплаты, но время сессии истекло.

5.6.3 Состояние APPROVED

Заказ, предавторизация по которому успешно завершилась в процессе совершения двухстадийного платежа, находится в состоянии Approved.

Двухстадийный платеж может быть отменен (Reversal) или завершен (Deposit).

Завершение двухстадийного платежа осуществляется с помощью операции Deposit.

Операция завершения оплаты двухстадийного платежа может быть проведена следующими способами:

- С помощью автоматизированных средств (запросы с использованием интерфейса REST/интерфейса на web-сервисах)
- С использованием кнопки «Завершить», которая доступна при просмотре детальной информации о транзакции из консоли.

При завершении оплаты заказа с использованием кнопки «Завершить» в открывшемся окне пользователем указывается сумма. Можно провести платеж как на полную сумму операции, так и на сумму, отличную от совершенной операции, но не более суммы оригинальной транзакции (Approve). Если указанная сумма депозита равна 0, операция Deposit выполняется на всю сумму оплаченной транзакции.

Операция Deposit может быть выполнена только один раз.

5.6.4 Состояние DEPOSITED

Заказ, оплата по которому была успешно завершена в процессе совершения одностадийного или двухстадийного платежа, находится в состоянии Deposited.

По заказу в состоянии Deposited можно провести возврат средств на сумму, не превышающую суммы заказа.

5.6.5 Состояние REVERSED

При одностадийных платежах отмена платежа возможна для заказов в состоянии Deposited. При двухстадийных платежах отмену можно выполнить для заказа в состоянии Approved.

Отмена платежа осуществляется от имени пользователя, у которого имеются соответствующие права.

Операция отмены может быть проведена следующими способами:

- С помощью автоматизированных средств (запросы с использованием интерфейса REST/интерфейса на web-сервисах).
- С использованием кнопки «Отменить», которая доступна при просмотре детальной информации о транзакции из консоли.

Операция отмены платежа может быть проведена до момента завершения платежа, но не позже срока, обозначенного в договоре на Интернет-эквайринг с Банком. При отмене операции происходит снятие блокировки с денежных средств на карте покупателя. Заказ при этом переходит в состояние Reversed.

Операцию отмены оплаты можно сделать только один раз. Если отмена платежа закончится ошибкой (т.е. запрос прошел через SmartVista и закончился ошибкой), то повторная операция отмены платежа не пройдет.

5.6.6 Состояние REFUND

Возврат средств осуществляется от имени пользователя системы, у которого существуют права на отмену операции.

Операция возврата средств может быть проведена следующими способами:

- С помощью автоматизированных средств (запросы с использованием интерфейса REST/интерфейса на web-сервисах).
- С использованием кнопки «Снять блокировку», которая доступна при просмотре детальной информации о транзакции из консоли.

Возврат средств (Refund) по заказу возможен более одного раза, до тех пор, пока общая сумма возврата не будет превышать сумму заказа. Сумма возврата указывается с учетом десятичных разрядов, т.е. в лунах, копейках, центах и пр.

5.7 История заказа

Кроме текущего состояния заказа в системе сохраняется история событий, которые происходили с заказом. В системе бывают следующие типы записей в истории:

Название события (анг.)	Название события (рус.)	Описание события
Order registered	Заказ создан	Заказ зарегистрирован в системе.
Authorization started	Авторизация начата	Начат процесс авторизации платежа.
Authorization finished	Авторизация завершена	Процесс авторизации платежа закончен.
Deposit started	Оплата начата	Начат процесс списания средств.
Deposit finished	Оплата завершена	Процесс списания средств завершен
Reversal started	Отмена начата	Начат процесс отмены платежа
Reversal finished	Отмена завершена	Процесс отмены платежа закончен
Refund started	Возврат начат	Начат процесс возмещения средств клиенту
Refund finished	Возврат завершен	Процесс возмещения средств клиенту закончен
Refund failed	Ошибка возврата	Запрос на возмещение средств клиенту закончился ошибкой (либо средства по этому заказу не были списаны, либо по техническим причинам (т.е. ошибки со стороны платежного шлюза или Банка))
Redirected to ACS	Перенаправление на ACS	В браузер пользователя передано перенаправление на ACS сервер

Returned from ACS	Возврат с ACS	Пользователь был перенаправлен с ACS на финальную страницу с результатами платежа
Redirected to merchant	Перенаправление на страницу продавца	В браузер пользователя передается перенаправление на сайт магазина
Payment attempted	Попытка оплаты	Начат процесс обработки параметров платежа, отправленных пользователем
Payment page loaded	Платежная страница загружена	Пользователю полностью отгрузилась платежная страница
Marked possible fraud	Помечено как возможный фрод	Заказ помечен пользователем как возможно фродовый
Failed 3D Secure	Неудачный 3D Secure	Directory Server или ACS ответили отказом на проверку карты по 3DS
SSL operation not allowed	SSL операции не разрешены	SSL авторизация по заказу запрещена настройками магазина
Session timeout occurred	Истекло время сессии	Истекло время, отведённое для оплаты

6 Алгоритм действий для подключения к платежному шлюзу

1. Получение логинов и паролей на тестовый сервер.
2. Разработка локального ПО, проведение тестовых платежей.
3. Верстка платежной страницы.*
4. Загрузка архива с платежной страницы на тестовый сервер.
5. Тестирование работоспособности платежной страницы.
6. Свяжитесь с банком, чтобы вашу платежную страницу проверили. Если проверка прошла успешно, сотрудники банка перенесут Вашу платежную страницу на боевой сервер.
7. Получение логинов и паролей на боевой сервер.
8. Переключение вашего магазина на использование промышленной системы.
9. Произведение тестовой оплаты настоящей картой (рекомендуется провести оплату по 3DS-карте, а также выполнить SSL-платеж).
10. Выполнить отмену и возврат платежа через личный кабинет платежа.
11. Подписание акта о готовности интернет-магазина.

* Шаги 3-6 могут являться необязательными, если банк предоставляет магазинам уже готовую платежную страницу. Детали необходимо уточнять у банка.

7 Реализация API через интерфейс REST

Взаимодействия реализуются как HTTP обращения методом POST на определенные URL, для каждого типа – свой. Параметры передаются как параметры POST запросов, значения должны быть совместимы с URL (т.е. url encoded).

Результат обработки запроса возвращается в виде JSON объекта. Например:

```
{"errorCode": "12", "errorMessage": "Empty amount"}
```

Для авторизации обращения магазина к системе платежного шлюза, в любом запросе со стороны магазина должны быть приведены имя и пароль магазина, которые представитель магазина ввел при регистрации магазина в системе. Значения имени и пароля передаются в следующих параметрах:

Название	Тип	Обязательность	Описание
userName	AN..30	да	Логин магазина, полученный при подключении
password	AN..30	да	Пароль магазина, полученный при подключении

В зависимости от выбранной схемы системы оплаты (одностадийная или двухстадийная) синтаксис запросов различается. Ниже описаны запросы для каждой из них.

Все текстовые поля должны иметь кодировку Юникод (UTF-8).

Спец-символы в REST запросе должны быть экранированы в соответствии с URL-кодированием. Таблица символов представлена здесь: <http://web-developer.name/urlcode/>. Например, пароль "qwe?rt%у" должен передаваться в виде "qwe%0Frt%25y".

Если Error code (Код ошибки) = 0, запрос был обработан Платежным шлюзом без системных ошибок (при этом error code не показывает статус заказа). Для получения статуса заказа следует использовать запрос getOrderStatus или getOrderStatusExtended.

7.1 Запросы, используемые при одностадийной оплате

7.1.1 Запрос регистрации заказа

Для регистрации заказа используйте запрос register (см.раздел [Координаты подключения](#)). Запрос позволяет зарегистрировать заказ в схеме приема платежа на стороне платежной системы.

Параметры запроса:

Название	Тип	Обязательно	Описание
userName	AN..30	да	Логин магазина, полученный при подключении
password	AN..30	да	Пароль магазина, полученный при подключении
orderNumber	AN..32	да	Номер (идентификатор) заказа в системе магазина
amount	N..20	да	Сумма платежа в минимальных единицах валюты
currency	N3	нет	Код валюты платежа ISO 4217. Если не указан, считается равным 643 (российские рубли).
returnUrl	AN..512	да	Адрес, на который надо перенаправить пользователя
description	AN..1024	нет	Описание заказа в свободной форме
language	A2	нет	Язык в кодировке ISO 639-1. Если не указан, будет использован язык, указанный в настройках магазина как язык по умолчанию (default language)
pageView	A..7	нет	В pageView передаётся признак – мобильное устройство (pageView=MOBILE) или десктоп (pageView=DESKTOP). В зависимости от этого в ответе будет ссылка на мобильную платёжную страницу или на обычную. Если параметр

			отсутствует, либо не соответствует этим значениям, то по умолчанию pageView=DESKTOP.
clientId	AN..255	нет	Номер (идентификатор) клиента в системе магазина. Используется для реализации функционала связок. Может присутствовать, если магазину разрешено создание связок и должен присутствовать, если заказ будет оплачен с помощью связки.
jsonParams	AN..1024	нет	Поля дополнительной информации для последующего хранения, вида {"param":value,"param2":value 2}. Данные поля могут быть переданы в процессинг банка для последующего отображения в реестрах.* Включение данного функционала возможно по согласованию с банком в период интеграции. При добавление параметра "FORCE_3DS2":"true" будет проверена возможность провести платеж через 3D Secure протокол версии 2.
sessionTimeoutSecs	N...9	нет	Продолжительность сессии в секундах. Не может превышать 1200 секунд. В случае если параметр не задан, будет использовано значение по умолчанию (1200 секунд – 20 минут).

* По умолчанию в процессинг банка передаются поля номер заказа orderNumber и его описание description (не более 99 символов, запрещены к использованию %, +, конец строки \r и перенос строки \n)

Параметры ответа:

Название	Тип	Обязательно	Описание
orderId	AN..64	нет	Номер заказа в платежной системе. Уникален в пределах системы. Отсутствует, если регистрация заказа на удалась по причине ошибки, детализированной в ErrorCode.
formUrl	AN..512	нет	URL платежной формы, на который надо перенаправить браузер клиента. Не возвращается если регистрация заказа не удалась по причине ошибки, детализированной в ErrorCode.
errorCode	N3	нет	Код ошибки. Может отсутствовать, если результат не привел к ошибке.
errorCodeString	AN3	нет	Поле errorCode в виде строки
error	boolean	нет	Поле показывающее наличие или отсутствие ошибки в обработке запроса. Равно true, если errorCode не равен нулю.
errorMessage	AN..512	нет	Описание ошибки на языке, переданном в параметре Language в запросе.

Коды ошибок (поле ErrorCode):

Значение	Описание
0	Обработка запроса прошла без системных ошибок
1	Заказ с таким номером уже зарегистрирован в системе
3	Неизвестная (запрещенная) валюта
4	Отсутствует обязательный параметр запроса
5	Ошибка значение параметра запроса
7	Системная ошибка

Расшифровка:

Значение	Описание
0	Обработка запроса прошла без системных ошибок
1	Заказ с таким номером уже обработан
3	Неизвестная валюта
4	Номер заказа не может быть пуст
4	Имя мерчанта не может быть пустым
4	Отсутствует сумма
4	URL возврата не может быть пуст
4	Пароль не может быть пуст
5	Неверная сумма
5	Неверный номер заказа
5	Неизвестное имя мерчанта
5	Доступ запрещён
5	Пользователь отключен
7	Системная ошибка

Например: В случае возвращения ошибки 4 (отсутствует обязательный параметр запроса) это означает, что в запросе на регистрацию заказа не хватает одного из следующих параметров: номер заказа, логин, сумма, URL возврата, пароль.

URL for API: <https://ipay.arca.am/payment/rest/register.do>

Пример запроса POST:

```
userName=Inst-api&password=111111&amount=100&currency=643&language=ru&orderNumber=G87654321&returnUrl=example.com&jsonParams={"orderNumber":G1234567890}&pageView=MOBILE
```

Пример ответа:

```
{"formUrl":"https://test.paymentgate.ru/ipaytest/merchants/Inst/mobile_payment_ru.htm1?mdOrder=32faa424-858a-4f22-92c5-a50a9cfe56dc","orderId":"32faa424-858a-4f22-92c5-a50a9cfe56dc"}
```

7.1.2 Запрос оплаты заказа

Для оплаты заказа используется запрос paymentorder. Запрос оплаты заказа подразумевает, что Магазином используется метод ввода Платежных реквизитов на страница Магазина. Данную операцию можно осуществлять, если есть соответствующие права в системе. Поддерживается только метод POST.

Параметры запроса:

Название	Тип	Обязательно	Описание
userName	AN..30	да	Логин магазина, полученный при подключении
password	AN..30	да	Пароль магазина, полученный при подключении
MDORDER	ANS36	да	Номер заказа, полученный при регистрации заказа
\$PAN	N..20	да	Номер карты
\$CVC	N..4	да	CVV2/CVC2
YYYY	N..4	да	Год
MM	N..2	да	Месяц
TEXT	A..512	да	Имя плательщика
language	A..2	да	Язык ответа
ip	AN..19	да	IP адрес плательщика
jsonParams	AN..1024	нет	Поля дополнительной информации для последующего хранения, вида {"param":value,"param2":value 2}. Данные поля могут быть переданы в процессинг банка для последующего отображения в реестрах.* Включение данного функционала возможно по согласованию с банком в период интеграции

Параметры ответа:

Название	Тип	Обязательно	Описание
errorCode	N3	да	Код ошибки. Может отсутствовать, если результат не привел к ошибке.
Error	AN..512	нет	Описание ошибки на языке.
info	AN..512	нет	<p>Результат попытки оплаты. Возможные значения представлены ниже:</p> <ul style="list-style-type: none"> • Ваш платёж обработан, происходит переадресация... • Операция отклонена. Проверьте введенные данные, достаточность средств на карте и повторите операцию. Происходит переадресация... • Извините, платеж не может быть совершен. Происходит переадресация... • Операция отклонена. Обратитесь в магазин. Происходит переадресация... • Операция отклонена. Обратитесь в банк, выпустивший карту. Происходит переадресация... • Операция невозможна. Аутентификация держателя карты завершена неуспешно. Происходит переадресация... • Нет связи с банком. Повторите позже. Происходит переадресация... • Истек срок ожидания ввода данных. Происходит переадресация... • Не получен ответ от банка. Повторите позже. Происходит переадресация...
redirect	AN..512	нет	Адрес возврата после оплаты
acsUrl	AN..512	Нет	Адрес Access Control server банка-эмитента, куда необходимо перенаправить браузер Пользователя для прохождения 3DSecure аутентификации. В SSL платежах не используется.
cReq	AN..512	Нет	BASE64 закодированный Challenge Request

Если errorCode = 0 и info = "Your order is proceeded, redirecting" ("Ваш платёж обработан, происходит переадресация...") - оплата прошла успешно. В других случаях с ошибкой, см. сообщение об ошибке errorMessage.

Коды ошибок (поле ErrorCode):

Значение	Описание
0	Обработка запроса прошла без системных ошибок

5	Ошибка значение параметра запроса
5	Использованы все попытки оплаты
5	Системная или внутренняя ошибка

Расшифровка:

Значение	Описание
0	Обработка запроса прошла без системных ошибок
5	Авторизация отклонена
5	Доступ запрещён
6	Неверный номер заказа
7	Недопустимая операция для текущего состояния заказа
7	Ошибка системы

Пример запроса POST:

```
userName=login&password=password&MDORDER=0d4b02cb-5147-4232-9012-4d38c743ahr6&
$PAN=5555555555555599&$CVC=123&YYYY=2015&MM=12&TEXT=Card Holder&language=ru
```

Пример ответа:

```
{"info": "Ваш платёж обработан, происходит переадресация...", "errorCode": 0}
```

Пример ответа во время 3DS 2.0:

```
{"info": "Ваш платёж обработан, происходит переадресация...", "acsUrl": "https://acs2.arca.am/acs/api/3ds2/creqbrw", "cReq": "BASE64 закодированный Challenge Request", "errorCode": 0}
```

Если платеж проходит с шагом проведения аутентификации по технологии 3Dsecure (в ответе на запрос присутствуют параметры acsUrl и paReq), то Магазин должен перенаправить клиента на ACS.

Для перенаправления на ACS, мерчант со своей стороны должен отправить клиента по адресу, указанному в параметре acsUrl с телом запроса cReq=creq, где

cReq –параметр из ответа на запрос оплаты,

redirect – параметр из ответа на запрос оплаты.

Запрос должен быть в формате POST.

ACS эмитента проводит аутентификацию держателя карты и перенаправляет клиента в платежный шлюз, передавая CRes. Платежный шлюз проводит авторизацию заказа при успешной аутентификации клиента на ACS, или отклоняет предавторизацию при неуспешной аутентификации клиента на ACS.

После завершения аутентификации держателя карты со стороны ACS плательщик перенаправляется на сайт Магазина по адресу, указанному в параметре redirect запроса paymentorder.

7.1.3 Запрос отмены оплаты заказа

Для отмены оплаты заказа используется запрос reverse. Данную операцию можно осуществлять, если есть соответствующие права в системе.

Параметры запроса:

Название	Тип	Обязательно	Описание
userName	AN..30	да	Логин магазина, полученный при подключении
password	AN..30	да	Пароль магазина, полученный при подключении
orderId	AN..64	да	Номер заказа в платежной системе. Уникален в пределах системы.

Параметры ответа:

Название	Тип	Обязательно	Описание
errorCode	N3	нет	Код ошибки. Может отсутствовать, если результат не привел к ошибке.
errorMessage	AN..512	нет	Описание ошибки на языке.

Коды ошибок (поле ErrorCode):

Значение	Описание
0	Обработка запроса прошла без системных ошибок
5	Ошибка значение параметра запроса
6	Незарегистрированный OrderId
7	Системная ошибка

Расшифровка:

Значение	Описание
0	Обработка запроса прошла без системных ошибок
5	Авторизация отклонена
5	Доступ запрещён
6	Неверный номер заказа
7	Недопустимая операция для текущего состояния заказа
7	Ошибка системы

URL for API: <https://ipay.arca.am/payment/rest/reverse.do>

Пример запроса POST:

```
amount=100&currency=643&language=ru&orderId=e5b59d3d-746b-4828-9da4-06f126e01b68&password=password&userName=login
```

Пример ответа:

```
{"errorCode":"7","errorMessage":"Недопустимая операция для текущего состояния заказа"}
```

7.1.4 Запрос возврата средств оплаты заказа

Для возврата средств оплаты заказа используется запрос refund. Данную операцию можно осуществлять, если есть соответствующие права в системе.

Параметры запроса:

Название	Тип	Обязательно	Описание
userName	AN..30	да	Логин магазина, полученный при подключении
password	AN..30	да	Пароль магазина, полученный при подключении
orderId	AN..64	да	Номер заказа в платежной системе. Уникален в пределах системы.
amount	N..20	да	Сумма платежа в минимальных единицах валюты

Параметры ответа:

Название	Тип	Обязательно	Описание
errorCode	N3	нет	Код ошибки. Может отсутствовать, если результат не привел к ошибке.
errorMessage	AN..512	нет	Описание ошибки на языке.

Коды ошибок (поле ErrorCode):

Значение	Описание
0	Обработка запроса прошла без системных ошибок
5	Ошибка значение параметра запроса
6	Незарегистрированный OrderId
7	Системная ошибка

Расшифровка:

Значение	Описание
5	Авторизация отклонена
5	Доступ запрещён
6	Неверный номер заказа
7	Платёж должен быть в корректном состоянии
7	Неверная сумма депозита (менее одного рубля)
7	Ошибка системы

URL for API: <https://ipay.arca.am/payment/rest/refund.do>

Пример запроса POST:

```
amount=500&currency=643&language=ru&orderId=5e97e3fd-1d20-4b4b-a542-f5995f5e8208&password=password&userName=login
```

Пример ответа:

```
{"errorCode":0}
```

7.1.5 Запрос состояния заказа

Для запроса состояния зарегистрированного заказа используется запрос getOrderStatusExtended.

Параметры запроса:

Название	Тип	Обязательно	Описание
userName	AN..30	да	Логин магазина, полученный при подключении
password	AN..30	да	Пароль магазина, полученный при подключении
orderId	AN..64	да*	Номер заказа в платежной системе. Уникален в пределах системы.
orderNumber	AN..32	да*	Номер (идентификатор) заказа в системе магазина.
language	A2	нет	Язык в кодировке ISO 639-1. Если не указан, считается, что язык – русский. Сообщение ошибке будет возвращено именно на этом языке.

* В запросе должен присутствовать либо orderId, либо orderNumber. Если в запросе присутствуют оба параметра, то приоритетным считается orderId.

Существует 3 набора параметров ответа. Какие именно наборы параметров будут возвращены, зависит от версии getOrderStatusExtended, указанной в настройках мерчанта.

Параметры ответа для версий 01, 02, 03: Эти параметры будут возвращены в ответе не зависимо от версии getOrderStatusExtended.

Название	Тип	Обязательно	Описание
orderNumber	AN..32	да	Номер (идентификатор) заказа в системе магазина
orderStatus	N2	нет	Состояние заказа в платежной системе. Значение выбирается из списка, приведенного ниже. Отсутствует, если заказ не был найден
actionCode	N3	да	Код ответа. Список возможных значений указан в разделе Приложение №1. Список значений action code.
actionCodeDescription	AN..512	да	Расшифровка кода ответа на языке, переданном в параметре Language в запросе
errorCode	N3	нет	Код ошибки.
errorMessage	AN..512	нет	Описание ошибки на языке, переданном в параметре Language в запросе
amount	N..20	да	Сумма платежа в копейках (или центах)
currency	N3	нет	Код валюты платежа ISO 4217. Если не указан, считается равным 643 (российские рубли)
date	ANS	да	Дата регистрации заказа
orderDescription	AN..512	нет	Описание заказа, переданное при его регистрации
ip	N..	да	IP адрес покупателя

Элемент cardAuthInfo (в элементе лежит структура, состоящая из списка элементов типа secureAuthInfo и атрибутов pan, expiration, cardholderName и approvalCode):			
pan	N..19	нет	Маскированный номер карты, которая использовалась для оплаты. Указан только после оплаты заказа
expiration	N6	нет	Срок истечения действия карты в формате YYYYMM. Указан только после оплаты заказа
cardholderName	A..64	нет	Имя держателя карты. Указан только после оплаты заказа
approvalCode	N6	нет	Код авторизации платежа. Указан только после оплаты заказа
Элемент secureAuthInfo (элемент состоит из элемента eci и элемента типа threeDSInfo, являющимся списком из cavv и xid):			
eci	N..4	нет	Электронный коммерческий индикатор. Указан только после оплаты заказа и в случае соответствующего разрешения. Возможные значения: 02 или 05 – платеж прошел с полной 3D Secure-аутентификацией 01 или 06 – 3D Secure-аутентификация прошла частично 07 – неудачная 3D Secure-аутентификация или же SSL-платеж
cavv	ANS..200	нет	Значение проверки аутентификации владельца карты. Указан только после оплаты заказа и в случае соответствующего разрешения.
xid	ANS..80	нет	Электронный коммерческий идентификатор транзакции. Указан только после оплаты заказа и в случае соответствующего разрешения
Элемент BindingInfo (элемент состоит из clientId и bindingId):			
clientId	AN..255	нет	Номер (идентификатор) клиента в системе магазина, переданный при регистрации заказа. Присутствует только если магазину разрешено создание связок

bindingId	AN..255	нет	Идентификатор связи созданной при оплате заказа или использованной для оплаты. Присутствует только если магазину разрешено создание связок
-----------	---------	-----	---

Параметры ответа для версий 02, 03:

Если для мерчанта указана версия getOrderStatusExtended 02 и выше, то в ответе, кроме описанных выше параметров, будут также возвращены следующие параметры.

Название	Тип	Обязательно	Описание
authDateTime	ANS	нет	Дата/время авторизации
authRefNum	AN..24	нет	Reference number
terminalId	AN..10	нет	Id терминала

Параметры ответа для версии 03: Если для мерчанта указана версия getOrderStatusExtended 03, то в ответе, кроме описанных выше параметров, будут также возвращены следующие параметры.

Название	Тип	Обязательно	Описание
Элемент paymentAmountInfo (состоит из approvedAmount, depositedAmount, refundedAmount и paymentState):			
approvedAmount	N..20	нет	Сумма, захолдированная на карте (используется только при двухстадийных платежах)
depositedAmount	N..20	нет	Сумма, подтвержденная для списания с карты
refundedAmount	N..20	нет	Сумма возврата
paymentState	N2	нет	Состояние заказа
Элемент bankInfo (состоит из bankName, bankCountryCode и bankCountryName):			
bankName	AN..200	нет	Наименование банка-эмитента
bankCountryCode	AN..4	нет	Код страны банка-эмитента
bankCountryName	AN..160	нет	Наименование страны банка-эмитента на языке, переданном в параметре language в запросе, или на языке пользователя, вызвавшего метод, если язык в запросе не указан

Поле OrderStatus может принимать следующие значения:

Номер состояния	Описание
0	Заказ зарегистрирован, но не оплачен
1	Деньги были захолдированы на карте (для двухстадийных платежей)
2	Проведена полная авторизация суммы заказа
3	Авторизация отменена
4	По транзакции была проведена операция возврата
5	Инициирована авторизация через ACS банка-эмитента
6	Авторизация отклонена

Коды ошибок (поле errorCode):

Значение	Описание
0	Обработка запроса прошла без системных ошибок
1	Заказ с таким номером уже зарегистрирован в системе
2	Заказ отклонен по причине ошибки в реквизитах платежа
3	Неизвестная (запрещенная) валюта
4	Отсутствует обязательный параметр запроса
5	Ошибка значение параметра запроса
6	Незарегистрированный OrderId
7	Системная ошибка

URL for API: <https://ipay.arca.am/payment/rest/getOrderStatusExtended.do>

Пример запроса POST:

```
userName=login&password=password&orderId=285b2973-4d02-4980-a54e-57c4d0d2xxx9&language=ru
```

Пример ответа:

```
{"attributes": [], "date": 1342007119386, "currency": "643", "amount": 100, "actionCode": 0, "orderNumber": "1212x31334z15", "orderDescription": "test", "orderStatus": 2, "ip": "217.12.97.50", "actionCodeDescription": "Платеж успешно обработан", "merchantOrderParams": [], "cardAuthInfo": {"expiration": "201512", "pan": "411111**1111", "approvalCode": "123456", "cardholderName": "dsd qdqd", "secureAuthInfo": {"eci": 5, "threeDSInfo": {"cavv": "AAABCpEAUBNCAHgBQAAAAAAA="}, "xid": "MDAwMDAwMDEzNDIwMDcxMTk3Njc="} } }
```

7.1.6 Запрос проверки вовлечённости карты в 3DS

Для проверки вовлечённости карты в 3DS используется запрос verifyEnrollment.

Параметры запроса:

Название	Тип	Обязательно	Описание
userName	AN..30	да	Логин пользователя (API)
password	AN..30	да	Пароль пользователя (API)
pan	N12...19	да	Номер карты

Параметры ответа:

Название	Тип	Обязательно	Описание
errorCode	N3	нет	Код ошибки. Может отсутствовать если результат не привел к ошибке.

errorMessage	AN..512	нет	Описание ошибки.
isEnrolled	A1	нет	Признак вовлечённости карты в 3DS. Возможные значения: Y, N, U.
emitterName	AN..160	нет	Наименование банка-эмитента.
emitterCountryCode	AN..4	нет	Код страны банка-эмитента.

Коды ошибок (поле ErrorCode):

Значение	Описание
0	Обработка запроса прошла без системных ошибок
1	Не указан номер карты
1	Номер карты должен быть числом, содержащим от 13 до 19 цифр
5	Пользователь должен изменить свой пароль.
5	Доступ запрещён (Пользователь должен иметь разрешение "Использовать Merchant API", мерчант должен иметь разрешение "Разрешено использовать сервис проверки вовлечённости в 3DS").
6	По заданному номеру карты информация не найдена.
7	Произошла системная ошибка.

Пример запроса POST:

```
userName=login&password=password&pan=4111111111111111
```

Пример ответа:

```
{"enrolled":"Y","emitterName":"TEST","emitterCountryCode":"RU","errorCode":"0","error Message":"Успешно"}
```

7.2 Запросы, используемые при двухстадийной оплате

7.2.1 Запрос регистрации заказа с предавторизацией

Для запроса регистрации заказа с предавторизацией используется запрос registerPreAuth.

Параметры запроса:

Название	Тип	Обязательно	Описание
userName	AN..30	да	Логин магазина, полученный при подключении
password	AN..30	да	Пароль магазина, полученный при подключении
orderNumber	AN..32	да	Номер (идентификатор) заказа в системе магазина
amount	N..20	да	Сумма платежа минимальных единицах валюты
currency	N3	нет	Код валюты платежа ISO 4217. Если не указан, считается равным 643 (российские рубли).
returnUrl	AN..512	да	Адрес, на который надо перенаправить пользователя
description	AN..1024	нет	Описание заказа в свободной форме
language	A2	нет	Язык в кодировке ISO 639-1. Если не указан, будет использован язык, указанный в настройках магазина как язык по умолчанию (default language)
pageView	A..7	нет	В pageView передаётся признак – мобильное устройство (pageView=MOBILE) или десктоп (pageView=DESKTOP). В зависимости от этого в ответе будет ссылка на мобильную платёжную страницу или на обычную. Если параметр отсутствует, либо не соответствует этим значениям, то по умолчанию pageView=DESKTOP.
clientId	AN..255	нет	Номер (идентификатор) клиента в системе магазина. Используется для реализации функционала связок. Может присутствовать, если магазину разрешено создание связок и должен присутствовать, если заказ будет оплачен с помощью связки.
jsonParams	AN..1024	нет	Поля дополнительной информации для последующего хранения, вида {"param":value,"param2":value 2}. Данные поля могут быть переданы в процессинг банка для последующего отображения в реестрах.* Включение данного функционала возможно по согласованию с банком в период интеграции.
sessionTimeoutSecs	N...9	нет	Продолжительность сессии в секундах. Не может превышать 1200 секунд. В случае если параметр не задан, будет использовано значение по умолчанию (1200 секунд – 20 минут).

* По умолчанию в процессинг банка передаются поля номер заказа orderNumber и его описание description (не более 99 символов, запрещены к использованию %, +, конец строки \r и перенос строки \n)

Параметры ответа:

Название	Тип	Обязательно	Описание
orderId	AN..64	нет	Номер заказа в платежной системе. Уникален в пределах системы. Отсутствует, если регистрация заказа не удалась по причине ошибки, детализированной в ErrorCode.
formUrl	AN..512	нет	URL платежной формы, на который надо перенаправить браузер клиента. Не возвращается если регистрация заказа не удалась по причине ошибки, детализированной в ErrorCode.
errorCode	N3	нет	Код ошибки. Может отсутствовать, если результат не привел к ошибке.
errorCodeString	AN3	нет	Поле errorCode в виде строки
error	boolean	нет	Поле показывающее наличие или отсутствие ошибки в обработке запроса. Равно true, если errorCode не равен нулю.
errorMessage	AN..512	нет	Описание ошибки на языке, переданном в параметре Language в запросе.

Коды ошибок (поле ErrorCode):

Значение	Описание
0	Обработка запроса прошла без системных ошибок
1	Заказ с таким номером уже зарегистрирован в системе
3	Неизвестная (запрещенная) валюта
4	Отсутствует обязательный параметр запроса
5	Ошибка значение параметра запроса
7	Системная ошибка

Расшифровка:

Значение	Описание
0	Обработка запроса прошла без системных ошибок
1	Заказ с таким номером уже обработан
3	Неизвестная валюта
4	Номер заказа не может быть пуст
4	Имя мерчанта не может быть пустым
4	Отсутствует сумма
4	URL возврата не может быть пуст
4	Пароль не может быть пуст
5	Неверная сумма
5	Неверный номер заказа
5	Неизвестное имя мерчанта
5	Доступ запрещён

5	Пользователь отключен
7	Системная ошибка

Например: В случае возвращения ошибки 4 (отсутствует обязательный параметр запроса) это означает, что в запросе на регистрацию заказа не хватает одного из следующих параметров: язык, номер заказа, логин, сумма, URL возврата, пароль.

<https://ipay.arca.am/payment/rest/registerPreAuth.do>

Пример запроса POST:

```
amount=100&currency=643&language=ru&orderNumber=87654321&password=password&returnUrl=
http%3A%2F%2Fyourserver%2Fresult&userName=login
&jsonParams={"orderNumber":"1234567890"}&pageView=DESKTOP
```

Пример ответа:

```
{"orderId":"61351fbd-ac25-484f-b930-4d0ce4101ab7","formUrl":
"https://test.paymentgate.ru/ipaytest/merchants/Inst/payment_ru.html?mdOrder=61351fbd
-ac25-484f-b930-4d0ce4101ab7"}
```

7.2.2 Запрос оплаты заказа

Для оплаты заказа используется запрос paymentorder. Запрос оплаты заказа подразумевает, что Магазином используется метод ввода Платежных реквизитов на страница Магазина. Данную операцию можно осуществлять, если есть соответствующие права в системе. Поддерживается только метод POST.

Параметры запроса:

Название	Тип	Обязательно	Описание
userName	AN..30	да	Логин магазина, полученный при подключении
password	AN..30	да	Пароль магазина, полученный при подключении
MDORDER	ANS36	да	Номер заказа, полученный при регистрации заказа
\$PAN	N..20	да	Номер карты
\$CVC	N..4	да	CVV2/CVC2
YYYY	N..4	да	Год
MM	N..2	да	Месяц
TEXT	A..512	да	Имя плательщика
language	A..2	да	Язык ответа
ip	AN..19	нет	IP адрес плательщика
jsonParams	AN..1024	нет	Поля дополнительной информации для последующего хранения, вида {"param":value,"param2":value 2}. Данные поля могут быть переданы в процессинг банка для последующего отображения в реестрах.* Включение данного функционала возможно по согласованию с банком в период интеграции

Параметры ответа:

Название	Тип	Обязательно	Описание
errorCode	N3	да	Код ошибки. Может отсутствовать, если результат не привел к ошибке.
Error	AN..512	нет	Описание ошибки на языке.
info	AN..512	нет	Результат попытки оплаты. Возможные значения представлены ниже: <ul style="list-style-type: none"> • Ваш платёж обработан, происходит переадресация... • Операция отклонена. Проверьте введенные данные, достаточность средств на карте и повторите операцию. Происходит переадресация... • Извините, платеж не может быть совершен. Происходит переадресация... • Операция отклонена. Обратитесь в магазин. Происходит переадресация... • Операция отклонена. Обратитесь в банк, выпустивший карту. Происходит переадресация...

			<ul style="list-style-type: none"> • Операция невозможна. Аутентификация держателя карты завершена неуспешно. Происходит переадресация... • Нет связи с банком. Повторите позже. Происходит переадресация... • Истек срок ожидания ввода данных. Происходит переадресация... • Не получен ответ от банка. Повторите позже. Происходит переадресация...
redirect	AN..512	нет	Адрес возврата после оплаты
acsUrl	AN..512	Нет	Адрес Access Control server банка-эмитента, куда необходимо перенаправить браузер Пользователя для прохождения 3DSecure аутентификации. В SSL платежах не используется.
paReq	AN..512	Нет	Адрес Access Control server банка-эмитента, куда необходимо перенаправить браузер Пользователя для прохождения 3DSecure аутентификации. В SSL платежах не используется.

Если errorCode = 0 и info = "Your order is proceeded, redirecting" ("Ваш платёж обработан, происходит переадресация...") - оплата прошла успешно. В других случаях с ошибкой, см. сообщение об ошибке errorMessage.

Коды ошибок (поле ErrorCode):

Значение	Описание
0	Обработка запроса прошла без системных ошибок
5	Ошибка значение параметра запроса
5	Использованы все попытки оплаты
5	Системная или внутренняя ошибка

Расшифровка:

Значение	Описание
0	Обработка запроса прошла без системных ошибок
5	Авторизация отклонена
5	Доступ запрещён
6	Неверный номер заказа
7	Недопустимая операция для текущего состояния заказа
7	Ошибка системы

Пример запроса POST:

```
userName=login&password=password&MDORDER=0d4b02cb-5147-4232-9012-4d38c743ahr6&
$PAN=5555555555555599&$CVC=123&YYYY=2015&MM=12&TEXT=Card Holder&language=ru
```

Пример ответа:

```
{"info":"Ваш платёж обработан, происходит переадресация","errorCode":0}
```

Если платеж проходит с шагом проведения аутентификации по технологии 3Dsecure (в ответе на запрос присутствуют параметры acsUrl и cReq), то Магазин должен перенаправить клиента на ACS.

Для перенаправления на ACS, мерчант со своей стороны должен отправить клиента по адресу, указанному в параметре acsUrl с телом запроса cReq=creq, где

cReq –параметр из ответа на запрос оплаты.

Запрос должен быть в формате POST.

ACS эмитента проводит аутентификацию держателя карты и перенаправляет клиента в платежный шлюз, передавая CRes. Платежный шлюз проводит авторизацию заказа при успешной аутентификации клиента на ACS, или отклоняет предавторизацию при неуспешной аутентификации клиента на ACS. После завершения аутентификации держателя карты со стороны ACS плательщик перенаправляется на сайт Магазина по адресу, указанному в параметре redirect запроса paymentorder.

7.2.3 Запрос завершения оплаты заказа

Для запроса завершения ранее пред авторизованного заказа используется запрос deposit.do. Данную операцию можно осуществлять, если есть соответствующие права в системе.

Параметры запроса:

Название	Тип	Обязательно	Описание
userName	AN..30	да	Логин магазина, полученный при подключении
password	AN..30	да	Пароль магазина, полученный при подключении
orderId	AN..64	да	Номер заказа в платежной системе. Уникален в пределах системы.
amount	N..20	да	Сумма платежа в минимальных единицах валюты

Параметры ответа:

Название	Тип	Обязательно	Описание
errorCode	N3	нет	Код ошибки. Может отсутствовать, если результат не привел к ошибке.
error	Boolean	Нет	Показывает наличие ошибки
errorMessage	AN..512	нет	Описание ошибки на языке.

Внимание!!! Если не указать параметр "amount", завершение произойдет на всю предавторизованную сумму.

Коды ошибок (поле ErrorCode):

Классификация:

Значение	Описание
0	Обработка запроса прошла без системных ошибок
5	Ошибка значение параметра запроса
6	Незарегистрированный OrderId
7	Системная ошибка

Расшифровка:

Значение	Описание
0	Обработка запроса прошла без системных ошибок
5	Авторизация отклонена
5	Пользователь отключен
5	У пользователя нет разрешения на депозит
5	Сумма депозита должна быть равной нулю или не менее одного рубля
6	Неверный номер заказа
7	Платёж должен быть в корректном состоянии
7	Ошибка системы

<https://ipay.arca.am/payment/rest/deposit.do>

Пример запроса POST:

```
amount=100&currency=643&language=ru&orderId=e5b59d3d-746b-4828-9da4-06f126e01b68&password=password&userName=login
```

Пример ответа:

```
{"errorCode":0}
```

7.2.4 Запрос отмены оплаты заказа

Для запроса отмены оплаты заказа используется запрос reverse. Данную операцию можно осуществлять, если есть соответствующие права в системе. Отмена платежа возможна только до конца того же дня, когда был выполнен платеж, т.е. к примеру если платеж выполнен в 23:50, то для его отмены есть только 10 минут. После смены суток платеж считается «выгруженным» для финансовой обработки и для его отмены необходимо уже обратится в банк.

Параметры запроса:

Название	Тип	Обязательно	Описание
userName	AN..30	да	Логин магазина, полученный при подключении
password	AN..30	да	Пароль магазина, полученный при подключении
orderId	AN..64	да	Номер заказа в платежной системе. Уникален в пределах системы.

Параметры ответа:

Название	Тип	Обязательно	Описание
errorCode	N3	нет	Код ошибки. Может отсутствовать, если результат не привел к ошибке.
error	boolean	нет	Поле показывающее наличие или отсутствие ошибки в обработке запроса. Равно true, если errorCode не равен нулю.
errorMessage	AN..512	нет	Описание ошибки на языке.

Коды ошибок (поле ErrorCode):

Классификация:

Значение	Описание
0	Обработка запроса прошла без системных ошибок
5	Ошибка значение параметра запроса
6	Незарегистрированный OrderId
7	Системная ошибка

Расшифровка:

Значение	Описание
0	Обработка запроса прошла без системных ошибок
5	Авторизация отклонена
5	Пользователь отключен
5	У пользователя нет разрешения на снятие блокировки средств
6	Неверный номер заказа
7	Платёж должен быть в корректном состоянии
7	Реверс невозможен. Суммы холдирования и депозита должны быть равны для транзакции после снятия блокировки средств.
7	Реверс невозможен. Причина: неверные внутренние значения, проверьте суммы холда, депозита
7	Ошибка системы

<https://ipay.arca.am/payment/rest/reverse.do>

Пример запроса POST:

```
amount=100&currency=643&language=ru&orderId=e5b59d3d-746b-4828-9da4-06f126e01b68&password=password&userName=login
```

Пример ответа:

```
{"errorCode":0}
```

7.2.5 Запрос возврата средств оплаты заказа

Для запроса возврата средств оплаты заказа используется запрос refund. Данную операцию можно осуществлять, если есть соответствующие права в системе.

Параметры запроса:

Название	Тип	Обязательно	Описание
userName	AN..30	да	Логин магазина, полученный при подключении
password	AN..30	да	Пароль магазина, полученный при подключении
orderId	AN..64	да	Номер заказа в платежной системе. Уникален в пределах системы.
Amount	N..20	да	Сумма платежа в минимальных единицах валюты

Параметры ответа:

Название	Тип	Обязательно	Описание
errorCode	N3	нет	Код ошибки. Может отсутствовать, если результат не привел к ошибке.
errorMessage	AN..512	нет	Описание ошибки на языке.
error	boolean	нет	Поле показывающее наличие или отсутствие ошибки в обработке запроса. Равно true, если errorCode не равен нулю.

Коды ошибок (поле ErrorCode):

Значение	Описание
0	Обработка запроса прошла без системных ошибок
5	Ошибка значение параметра запроса
6	Незарегистрированный OrderId
7	Системная ошибка

Расшифровка:

Значение	Описание
0	Обработка запроса прошла без системных ошибок
5	Авторизация отклонена
5	Пользователь отключен
5	У пользователя нет разрешения на снятие блокировки средств
6	Неверный номер заказа
7	Платёж должен быть в корректном состоянии
7	Неверная сумма депозита (менее одного рубля)
7	Ошибка системы

<https://ipay.arca.am/payment/rest/refund.do>:

Пример запроса POST:

```
amount=500&currency=643&language=ru&orderId=5e97e3fd-1d20-4b4b-a542-f5995f5e8208&password=password&userName=login
```

Пример ответа:

```
{"errorCode":0}
```

7.2.6 Запрос состояния заказа

Для запроса состояния зарегистрированного заказа используется запрос `getOrderStatus.do`

Параметры запроса:

Название	Тип	Обязательно	Описание
userName	AN..30	да	Логин магазина, полученный при подключении
orderId	AN..64	да*	Номер заказа в платежной системе. Уникален в пределах системы.
password	AN..30	да	Пароль магазина, полученный при подключении
language	A2	нет	Язык в кодировке ISO 639-1. Если не указан, считается, что язык – русский. Сообщение ошибке будет возвращено именно на этом языке.

Параметры ответа

Название	Тип	Обязательно	Описание
orderNumber	AN..32	да	Номер (идентификатор) заказа в системе магазина
orderStatus	N2	нет	Состояние заказа в платежной системе. Значение выбирается из списка, приведенного ниже. Отсутствует, если заказ не был найден
errorCode	N3	нет	Код ошибки.
errorMessage	AN..512	нет	Описание ошибки на языке, переданном в параметре <code>Language</code> в запросе
amount	N..20	да	Сумма платежа в копейках (или центах)
depositAmount	N..20	Да	Сумма завершенного платежа в центах или копейках
approvalCode	AN6	Да	Код авторизации
authCode	N2	Да	Код завершения платежа
currency	N3	нет	Код валюты платежа ISO 4217. Если не указан, считается равным 643 (российские рубли)
ip	AN..100	да	IP адрес покупателя
cardholderName	AN..200	Нет	Имя держателя карты
svfeResponse	N3	Да	Код ответа SVFE
Элемент Params		нет	В элементе содержится структура, состоящая из списка элементов типа пара Имя-Значение с параметрами заказа предназначенными для выдачи магазину.
error	Boolean	нет	Поле показывающее наличие или отсутствие ошибки в обработке запроса. Равно true, если errorCode не равен нулю.
pan	N..19	нет	Маскированный номер карты, которая использовалась для оплаты. Указан только после оплаты заказа
expiration	N6	нет	Срок истечения действия карты в формате

7.2.7 Расширенный запрос состояния заказа

Для расширенного запроса состояния зарегистрированного заказа используется запрос `getOrderStatusExtended`.

Параметры запроса:

Название	Тип	Обязательно	Описание
userName	AN..30	да	Логин магазина, полученный при подключении
password	AN..30	да	Пароль магазина, полученный при подключении
orderId	AN..64	да*	Номер заказа в платежной системе. Уникален в пределах системы.
orderNumber	AN..32	да*	Номер (идентификатор) заказа в системе магазина.
language	A2	нет	Язык в кодировке ISO 639-1. Если не указан, считается, что язык – русский. Сообщение ошибке будет возвращено именно на этом языке.

* В запросе должен присутствовать либо orderId, либо orderNumber. Если в запросе присутствуют оба параметра, то приоритетным считается orderId.

Существует 3 набора параметров ответа. Какие именно наборы параметров будут возвращены, зависит от версии `getOrderStatusExtended`, указанной в настройках мерчанта.

Параметры ответа для версий 01, 02, 03:

Эти параметры будут возвращены в ответе независимо от версии `getOrderStatusExtended`.

Название	Тип	Обязательно	Описание
orderNumber	AN..32	да	Номер (идентификатор) заказа в системе магазина
orderStatus	N2	нет	Состояние заказа в платежной системе. Значение выбирается из списка, приведенного ниже. Отсутствует, если заказ не был найден
actionCode	N3	Да	Код ответа. Список возможных значений указан в разделе Приложение №1. Список значений action code .
actionCodeDescription	AN..512	да	Расшифровка кода ответа на языке, переданном в параметре Language в запросе
errorCode	N3	нет	Код ошибки.
errorMessage	AN..512	нет	Описание ошибки на языке, переданном в параметре Language в запросе
amount	N..20	да	Сумма платежа в копейках (или центах)
currency	N3	нет	Код валюты платежа ISO 4217. Если не указан, считается равным 643 (российские рубли)
date	ANS	да	Дата регистрации заказа
orderDescription	AN..512	нет	Описание заказа, переданное при его регистрации
ip	N..	да	IP адрес покупателя
Элемент cardAuthInfo			В элементе лежит структура, состоящая из списка элементов типа secureAuthInfo и атрибутов pan, expiration, cardholderName и approvalCode

Элемент merchantOrderParams		нет	В элементе содержится структура, состоящая из списка элементов типа пара Имя-Значение с параметрами относящимися к заказу. Например, для рекуррентного платежа структура может выглядеть так: [{"name":"recurringId","value":"ad49eade-d0cd-496d-8e38-1b12eb580402"}, {"name":"recurringInitializeTransactionId","value":"00001608813142380000"}, {"name":"recurringInitializeTransactionRN","value":"987654321012"}]
Элемент attributes		нет	В элементе содержится структура, состоящая из списка элементов типа пара Имя-Значение с атрибутами относящимися к заказу. Например, атрибут mdOrder: [{"name":"mdOrder","value":"867252b9-53c9-42cc-bb72-7fec49d7ff16"}]
Элемент refunds		нет	В элементе содержится структура, состоящая из списка элементов типа пара Ключ-Значение с возвратами относящимися к заказу, если они были. Например, [{"referenceNumber":"001017626432","actionCode":"000","amount":500,"approvalCode":"626433","date":"20220309134437"}]
error	boolean	нет	Поле показывающее наличие или отсутствие ошибки в обработке запроса. Равно true, если errorCode не равен нулю.
pan	N..19	нет	Маскированный номер карты, которая использовалась для оплаты. Указан только после оплаты заказа
expiration	N6	нет	Срок истечения действия карты в формате

			YYYYMM. Указан только после оплаты заказа
cardholderName	A..64	нет	Имя держателя карты. Указан только после оплаты заказа
approvalCode	N6	нет	Код авторизации платежа. Указан только после оплаты заказа
Элемент secureAuthInfo (элемент состоит из элемента eci и элемента типа threeDSInfo, являющимся списком из cavv и xid):			
eci	N..4	нет	Электронный коммерческий индикатор. Указан только после оплаты заказа и в случае соответствующего разрешения
cavv	ANS..200	нет	Значение проверки аутентификации владельца карты. Указан только после оплаты заказа и в случае соответствующего разрешения
xid	ANS..80	нет	Электронный коммерческий идентификатор транзакции. Указан только после оплаты заказа и в случае соответствующего разрешения
Элемент BindingInfo (элемент состоит из clientId и bindingId):			
clientId	AN..255	нет	Номер (идентификатор) клиента в системе магазина, переданный при регистрации заказа. Присутствует только если магазину разрешено создание связок
bindingId	AN..255	нет	Идентификатор связи созданной при оплате заказа или использованной для оплаты. Присутствует только если магазину разрешено создание связок

Параметры ответа для версий 02, 03:

Если для мерчанта указана версия getOrderStatusExtended 02 и выше, то в ответе, кроме описанных выше параметров, будут также возвращены следующие параметры.

Название	Тип	Обязательно	Описание
authDateTime	ANS	нет	Дата/время авторизации
authRefNum	AN..24	нет	Reference number
terminalId	AN..10	нет	Id терминала

Параметры ответа для версии 03:

Если для мерчанта указана версия getOrderStatusExtended 03, то в ответе, кроме описанных выше параметров, будут также возвращены следующие параметры.

Название	Тип	Обязательно	Описание
Элемент paymentAmountInfo (состоит из approvedAmount, depositedAmount, refundedAmount и paymentState):			
approvedAmount	N..20	нет	Сумма, захолдированная на карте (используется только при двухстадийных платежах)
depositedAmount	N..20	нет	Сумма, подтвержденная для списания с карты
refundedAmount	N..20	нет	Сумма возврата
paymentState	N2	нет	Состояние заказа
Элемент bankInfo (состоит из bankName, bankCountryCode и bankCountryName):			
bankName	AN..200	нет	Наименование банка-эмитента
bankCountryCode	AN..4	нет	Код страны банка-эмитента
bankCountryName	AN..160	нет	Наименование страны банка-эмитента на языке, переданном в параметре language в запросе, или на языке пользователя, вызвавшего метод, если язык в запросе не указан

Поле OrderStatus может принимать следующие значения:

Номер состояния	Описание
0	Заказ зарегистрирован, но не оплачен
1	Деньги были захолдированы на карте (для двухстадийных платежей)
2	Проведена полная авторизация суммы заказа
3	Авторизация отменена
4	По транзакции была проведена операция возврата
5	Инициирована авторизация через ACS банка-эмитента
6	Авторизация отклонена

Коды ошибок (поле errorCode):

Значение	Описание
0	Обработка запроса прошла без системных ошибок
1	Заказ с таким номером уже зарегистрирован в системе
2	Заказ отклонен по причине ошибки в реквизитах платежа
3	Неизвестная (запрещенная) валюта
4	Отсутствует обязательный параметр запроса
5	Ошибка значение параметра запроса
6	Незарегистрированный OrderId
7	Системная ошибка

[https://ipay.arca.am/payment/rest/getOrderStatusExtended.do:](https://ipay.arca.am/payment/rest/getOrderStatusExtended.do)

Пример запроса POST:

```
userName=login&password=password&orderId=285b2973-4d02-4980-a54e-57c4d0d2xxx9&language=ru
```

Пример ответа:

```
{"attributes": [], "date": 1342007119386, "currency": "643", "amount": 100, "actionCode": 0, "orderNumber": "1212x31334z15", "orderDescription": "test", "orderStatus": 2, "ip": "217.12.97.50", "actionCodeDescription": "Платеж успешно обработан", "merchantOrderParams": [], "cardAuthInfo": {"expiration": "201512", "pan": "411111**1111", "approvalCode": "123456", "cardholderName": "dsd qdqd", "secureAuthInfo": {"eci": 5, "threeDSInfo": {"cavv": "AAABCpEAUBNCAHegBQAAAAAAA=", "xid": "MDAwMDAwMDEzNDIwMDcxMTk3Njcs="}}}}, {"status": "Success", "message": "Success"}]
```

7.2.8 Запрос проверки вовлечённости карты в 3DS

Для проверки вовлечённости карты в 3DS используется запрос verifyEnrollment.

Параметры запроса:

Название	Тип	Обязательно	Описание
userName	AN..30	да	Логин пользователя (API)
password	AN..30	да	Пароль пользователя (API)
pan	N12...19	да	Номер карты

Параметры ответа:

Название	Тип	Обязательно	Описание
errorCode	N3	нет	Код ошибки. Может отсутствовать если результат не привел к ошибке.
errorMessage	AN..512	нет	Описание ошибки.
enrolled	A1	нет	Признак вовлечённости карты в 3DS. Возможные значения: Y, N, U.
emitterName	AN..160	нет	Наименование банка-эмитента.
emitterCountryCode	AN..4	нет	Код страны банка-эмитента.
error	boolean	нет	Показывает наличие ошибки при обработке запроса

Коды ошибок (поле errorCode):

Значение	Описание
0	Обработка запроса прошла без системных ошибок
1	Не указан номер карты
1	Номер карты должен быть числом, содержащим от 13 до 19 цифр
5	Пользователь должен изменить свой пароль.
5	Доступ запрещён (Пользователь должен иметь разрешение "Использовать Merchant API", мерчант должен иметь разрешение "Разрешено использовать сервис проверки вовлечённости в 3DS").
6	По заданному номеру карты информация не найдена.
7	Произошла системная ошибка.

[https://ipay.arca.am/payment/rest/verifyEnrollment.do:](https://ipay.arca.am/payment/rest/verifyEnrollment.do)

Пример запроса POST:

```
userName=login&password=password&pan=4111111111111111
```

Пример ответа:

```
{"enrolled":"Y","emitterName":"TEST","emitterCountryCode":"RU","errorCode":"0","error  
Message":"Успешно"}
```

8 Оформление платежной страницы

ВНИМАНИЕ! Перед тем как ознакомиться с данной главой и приступить к созданию платежной страницы рекомендуется обсудить необходимость данного шага с банком-партнером. Возможно банк предоставит готовую платежную страницу. При использовании метода с вводом Платежных реквизитов на стороне Магазина платежная страница также не требуется.

Файл с платежными страницами должен представлять из себя архив zip-формата, где в корне находятся страницы и папки с js-скриптами, css-стилями и картинками. Пример архива с платежными страницами может быть предоставлен банком.

Для архивации рекомендуется использовать архиватор 7-zip.

8.1 Требования к коду страницы платёжного интерфейса.

Страница должна представлять из себя xhtml-страницу с обязательным объявлением DTD.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Запрещено использовать абсолютные адреса для подключения каких-либо ресурсов (картинки, скрипты, стили). Все адреса должны быть относительны расположения страницы и папок, где размещены все необходимые ресурсы (<http://host/images/test.jpg> - нельзя, <images/test.jpg> - можно).

*Внимание!!!!!! *Использование объявленного стандарта XHTML - <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" " [http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">](http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd) обязательно, в противном случае возможна неработоспособность в части браузеров.

Необходимые страницы для платёжного интерфейса:

- `payment_{locale}.html` – платёжная страница;
- `errors_{locale}.html` – страница ошибок. Где `locale` - язык страницы в кодировке ISO 639-1, например `ru` для русского или `en` для английского.

Если также предполагается использование страниц для мобильных устройств, потребуются страницы:

- `mobile_payment_{locale}.html` – платёжная страница;
- `mobile_errors_{locale}.html` – страница ошибок.

Где `locale` - язык страницы в кодировке ISO 639-1, например `ru` для русского или `en` для английского.

Какие именно страницы, обычные или мобильные, должны загружаться, определяется специальным параметром `pageView` при регистрации заказа (см. разделы [Запрос регистрации заказа](#) и [Запрос регистрации заказа с предавторизацией](#))

При верстке страниц (платежная страница, страница ошибок) нужно использовать кодировку Юникод (UTF-8).

8.2 Требования к платёжной странице.

Страница должна содержать ряд необходимых объектов, а также ряд полей для ввода платёжной информации с определённым названием.

8.2.1 Название страницы

Название обычной страницы – payment_<ln>.html, Название страницы для мобильного устройства -mobile_payment_<ln>.html, где <ln> - двухбуквенное обозначение локали страницы в кодировке ISO 639-1 (например, ru – русский, en - английский).

8.2.2 Заголовок страницы

В заголовке страницы должны подключаться следующие скрипты:

Стандартный вариант:

```
<script type="text/javascript" src="../../js/jquery-1.8.1.min.js"></script>
<script type="text/javascript" src="../../js/jquery-ui-1.9.2.custom.min.js">
</script>
<script type="text/javascript" src="../../js/jquery.timers-1.2.js"></script>
<script type="text/javascript" src="../../js/jquery.url.js"></script>
<script type="text/javascript" src="../../js/jquery.payment_new.js"></script>
<script type="text/javascript" src="../../js/maskedinput.js"></script>
<script type="text/javascript" src="../../js/jquery.main.js"></script>
<script type="text/javascript" src="../../js/select.js"></script>
<script type="text/javascript" src="../../js/localization.js"></script>
<script>
$(document).payment({
});
</script>
```

Расширенный вариант:

```
<script type="text/javascript" src="../../js/jquery-1.8.1.min.js"></script>
<script type="text/javascript" src="../../js/jquery-ui-1.9.2.custom.min.js">
</script>
<script type="text/javascript" src="../../js/jquery.timers-1.2.js"></script>
<script type="text/javascript" src="../../js/jquery.url.js"></script>
<script type="text/javascript" src="../../js/jquery.payment_new.js"></script>
<script type="text/javascript" src="../../js/maskedinput.js"></script>
<script type="text/javascript" src="../../js/jquery.main.js"></script>
<script type="text/javascript" src="../../js/select.js"></script>
<script type="text/javascript" src="../../js/localization.js"></script>
<script>
$(document).payment({
language: "ru", messageAjaxError: "Сервис временно недоступен. Попробуйте позднее.",
messageTimeRemaining: "До окончания сессии осталось #MIN#: #SEC#",
}); </script>
```

Поля расширенного вида скрипта должны быть заполнены следующим образом:

- language – значение название языка, совпадающее с выбранным для названия страницы
- messageAjaxError – сообщение о внутренней ошибки Ajax (возникает например при отсутствии доступа к системе)
- messageTimeRemaining – сообщение счётчика сессии. В нём обязательно должны быть указаны ключевые слова «#MIN#» и «#SEC#», которые в реальном времени будут заменять на минуты и секунды, обозначающие время до окончания сессии.

8.2.3 Тело страницы

Все блоки и элементы, описанные ниже в данном параграфе, обязательно должны быть размещены в теле страницы, если явно не указано иное.

```
<div id="orderNumber"></div>
```

блок, где содержится уникальный номер заказа

```
<div id="amount"></div>
```

блок, где содержится сумма оплаты заказа

```
<div id="description"></div>
```

блок, где содержится описание заказа

8.2.4 Платежная форма

Страница должна содержать платежную форму:

```
<form name="PaymentForm" action="#" method="post" id="formPayment">
<input type="hidden" id="expiry" >
<input type="hidden" id="mdOrder" >
<input type="hidden" id="location" value="/.../" >
<input type="hidden" id="language" value=<ln> > </form>
```

Все указанные выше hidden-поля обязательны. Значение поля language должно содержать двухбуквенное обозначение локали страницы.

Форма также должна содержать поля для ввода информации для проведения платежа:

- Поле для ввода номера кредитной карты

```
<input name="$PAN" id="iPAN" maxlength="19" type="text" autocomplete="off" />
```

- Селектор месяца и селектор года (заполняется автоматически при загрузке страницы) истечения срока действия кредитной карты

```

<select name="MM" id="month">
<option value="01" selected> 1 - январь</option>
<option value="02"> 2 - февраль</option>
<option value="03"> 3 - март</option>
<option value="04"> 4 - апрель</option>
<option value="05"> 5 - май</option>

<option value="06"> 6 - июнь</option>
<option value="07"> 7 - июль</option>
<option value="08"> 8 - август</option>
<option value="09"> 9 - сентябрь</option>
<option value="10">10 - октябрь</option>
<option value="11">11 - ноябрь</option>
<option value="12">12 - декабрь</option>
</select> /
<select name="YYYY" id="year">
</select>

```

- Поле ввода имени владельца карты (Cardholder name)

```
<input name="TEXT" id="iTEXT" maxlength="90" type="text" autocomplete="off" />
```

- Поле ввода cvc/cvv/cid -кода

```
<input name="$CVC" id="iCVC" maxlength="3" type="password" autocomplete="off" />
```

- Кнопка подтверждения оплаты

```
<input value="Оплатить" type="button" id="buttonPayment">
```

После формы оплаты, ниже должен быть размещён следующие код:

```

<form id="acs" method="post" action="">
    <input type="hidden" id="MD" name="MD"/>
    <input type="hidden" id="PaReq" name="PaReq"/>
    <input type="hidden" id="TermUrl" name="TermUrl"/>
</form>
<form id="threeDS2" method="post" action="">
    <input type="hidden" id="CReq" name="creq"/>
</form>
<div id="threeDS2MethodDiv" style="visibility: hidden;"></div>

```

На странице оплаты должны быть также размещены следующие объекты:

- блок, где отображаются ошибки (например, неверные данные по карте)

```
<div id="errorBlock" style="color:red;"></div>
```

- блок, где отображается сообщение о том, сколько ещё времени до конца сессии оплаты.

```
<div id="numberCountdown"></div>
```

- блок, где отображается информационное сообщение при переходе со страницы оплаты на

итоговую страницу.

```
<div id="infoBlock"></div>
```

- блок, где отображается индикатора прогресса выполнения запроса к серверу (при подтверждении оплаты и последующему обращению к серверу)

```
<div id="indicator" style="display:none;"></div>
```

Если предполагается использование IFrame, то в тело страницы необходимо добавить блок:

```
<iframe name="iframe_name" id="iframe_id" src="formUrl" style="width: 100%; height: 700px; border: 0 none;" scrolling="no" frameborder="0"></iframe>
```

При выполнении всех требований на платежной странице при оплате заказа будут отображаться:

- сумма заказа
- номер заказа в системе магазина
- описание заказа (отображается только при заполнении поля description)

8.3 Требования к странице ошибок

Страница должна содержать ряд необходимых объектов.

8.3.1 Название страницы

Название обычной страницы – errors_<ln>.html,

Название страницы для мобильного устройства – mobile_errors_<ln>.html,

где <ln> - двухбуквенное обозначение локали страницы (например, ru – русский, en - английский, ISO 639-1).

8.3.2 Заголовок страницы

В заголовке страницы должны подключаться следующие скрипты:

```
<script type="text/javascript" src="../../js/jquery-1.8.1.min.js"></script>
<script type="text/javascript" src="../../js/jquery.timers-1.2.js"></script>
<script type="text/javascript" src="../../js/jquery.url.js"></script>
<script type="text/javascript" src="../../js/error_page.js"></script>
```

```
<script type="text/javascript">
var lang = "<ln>"; </script>
```

где <ln> - двухбуквенное обозначение локали страницы (например, ru – русский, en - английский, ISO 639-1).

8.3.3 Тело страницы

Все блоки и элементы, описанные ниже в данном параграфе, обязательно должны быть размещены в теле страницы. Форма:

```
<form name="errorForm" action="#" method="post" id="errorForm">
<input type="hidden" id="language" value="" type="hidden">
</form>
```

где - двухбуквенное обозначение локали страницы (например, ru – русский, en - английский, ISO 639-1).

Форма также должна содержать блок, где отображаются ошибки (например, об окончании сессии или сообщение о том, что оплата уже была совершена).

```
<div id="errorBlock" style="color:red;"></div>
```

9 Координаты подключения

При регистрации мерчанта, представителю магазина предоставляется:

- пара логин+пароль, который можно использовать в личном кабинете
- пара логин+пароль, которую нужно использовать в протоколах
- в том случае, если магазин использует платежи по связкам – еще одна пара логин+пароль, которую нужно использовать в протоколах

Параметры, которые предоставляются для работы в тестовой среде, не могут быть использованы для работы в боевом режиме.

URL для доступа к личному кабинету в тестовой системе:

<http://91.199.226.7:8444/payment/admin>

URL для доступа к методам REST в тестовой системе:

Название метода	URL
Регистрация заказа	https://ipaytest.arca.am:8445/payment/rest/register.do
Регистрация заказа с предавторизацией	https://ipaytest.arca.am:8445/payment/rest/registerPreAuth.do
Запрос проведения платежа при использовании метода ввода платежных реквизитов на стороне Магазина	https://ipaytest.arca.am:8445/payment/rest/paymentorder.do
Запрос завершения оплаты заказа	https://ipaytest.arca.am:8445/payment/rest/deposit.do
Запрос отмены оплаты заказа	https://ipaytest.arca.am:8445/payment/rest/reverse.do
Запрос возврата средств оплаты заказа	https://ipaytest.arca.am:8445/payment/rest/refund.do
Получение статуса заказа	https://ipaytest.arca.am:8445/payment/rest/getOrderStatus.do
Получение расширенного статуса заказа	https://ipaytest.arca.am:8445/payment/rest/getOrderStatusExtended.do
Запрос проверки вовлеченности карты в 3DS	https://ipaytest.arca.am:8445/payment/rest/verifyEnrollment.do
Запрос проведения оплаты по связкам	https://ipaytest.arca.am:8445/payment/rest/paymentOrderBinding.do
Запрос деактивации связки	https://ipaytest.arca.am:8445/payment/rest/unBindCard.do
Запрос активации связки	https://ipaytest.arca.am:8445/payment/rest/bindCard.do
Запрос списка возможных связок для мерчанта	https://ipaytest.arca.am:8445/payment/rest/getBindings.do

URL для доступа к личному кабинету в боевой системе:

<https://ipay.arca.am/payment/admin>

URL для доступа к методам REST в боевой системе:

Название метода	URL
Регистрация заказа	https://ipay.arca.am/payment/rest/register.do
Регистрация заказа с предавторизацией	https://ipay.arca.am/payment/rest/registerPreAuth.do
Запрос завершения оплаты заказа	https://ipay.arca.am/payment/rest/deposit.do
Запрос проведения платежа при использовании метода ввода платежных реквизитов на стороне Магазина	https://ipay.arca.am/payment/rest/paymentorder.do
Запрос отмены оплаты заказа	https://ipay.arca.am/payment/rest/reverse.do
Запрос возврата средств оплаты заказа	https://ipay.arca.am/payment/rest/refund.do
Получение статуса заказа	https://ipay.arca.am/payment/rest/getOrderStatus.do
Получение расширенного статуса заказа	https://ipay.arca.am/payment/rest/getOrderStatusExtended.do
Запрос проверки вовлеченности карты в 3DS	https://ipay.arca.am/payment/rest/verifyEnrollment.do
Запрос проведения оплаты по связкам	https://ipay.arca.am/payment/rest/paymentOrderBinding.do
Запрос деактивации связки	https://ipay.arca.am/payment/rest/unBindCard.do
Запрос активации связки	https://ipay.arca.am/payment/rest/bindCard.do
Запрос списка возможных связок для мерчанта	https://ipay.arca.am/payment/rest/getBindings.do

10 Тестовые карты

Данные тестовых карт для проведения платежей необходимо получать от банка-партнера. См. пункт [Алгоритм действий для подключения к платежному шлюзу](#).

11 Описание функционала связок

11.1 Общее описание

Данный функционал используется для привязки платежных реквизитов карты к уникальному идентификатору покупателя в системе магазина (например, к логину, адресу электронной почты, номеру телефона и пр.). Основной смысл использования связок состоит в том, что при использовании связки Плательщик более не должен нигде вводить платежные реквизиты карты для осуществления платежа. Более того, механизм связок позволяет Магазину осуществлять платежи без присутствия или вмешательства Плательщика, что может быть использовано к примеру для периодических платежей, подписок и проч. Для создания связки Плательщику необходимо провести один успешный «обычный» платеж, в результате которого и создается связка.

В целях безопасности и защиты данных Плательщиков данные по связкам, в частности, платежные реквизиты карты, хранятся на сервере Платежного шлюза. Магазину передается только уникальный идентификатор связки, который и может быть использован при последующих платежах.

11.2 Технические особенности

Исходя из архитектуры системы для использования связок мерчанту необходимо получить и использовать 2 логина вместо одного:

- первый (основной логин) используется для проведения обычных платежей и для создания связок
- второй логин используется только для проведения платежей по связкам, сгенерированным с помощью первого логина

11.3 Запросы по связкам через интерфейс REST

11.3.1 Создание связки

Для создания связки Плательщику необходимо провести один успешный «обычный» платеж. Именно во время успешного платежа проверяются платежные реквизиты карты, в результате чего их можно привязать к идентификатору связки. Также при этом должны быть соблюдены следующие условия:

- Магазину разрешено использование механизма связок
- Магазин передал параметр clientId во время регистрации заказа

После проведения успешного платежа Магазин может получить идентификатор новой связки отправив запрос состояния заказа (см. пункт [Запрос состояния заказа](#)). В числе прочих параметров, в ответе содержится элемент BindingInfo, в свою очередь содержащий параметр bindingId.

11.3.2 Запрос проведения платежа по связкам

Для проведения платежа по связкам используется запрос paymentOrderBinding (см. раздел [Координаты подключения](#)). Предварительно заказ должен быть создан вызовом метода register (см. [Запрос регистрации заказа](#)) или registerPreAuth ([Запрос регистрации заказа с предавторизацией](#)) при этом в запросе должен быть указан параметр clientId.

Параметры запроса:

Название	Тип	Обязательно	Описание
userName	AN..30	да	Логин магазина, полученный при подключении. ВАЖНО. Должен использоваться тот логин, который предназначен для проведения платежей по связкам. См. пункт Подключение
Password	AN..30	да	Пароль магазина, полученный при подключении
mdOrder	AN..64	да	Номер заказа в платежной системе. Уникален в пределах системы.
bindingId	AN..255	да	Идентификатор связки созданной при оплате заказа или использованной для оплаты.

			Присутствует только если магазину разрешено создание связок.
Cvc	N..3	нет	CVC код
Language	A2	нет	Язык в кодировке ISO 639-1. Если не указан, будет использован язык, указанный в настройках магазина как язык по умолчанию (default language)

Параметры ответа:

Название	Тип	Обязательно	Описание
redirect	ANS..*	(при SSL платеже)	URL, на который производится переадресация после платежа
Info	ANS..*	да	Информационное сообщение
success	N1	да	Код ошибки
errorCode	N3	да	Код ошибки. Может отсутствовать если результат не привел к ошибке.
error	AN..*	(при ошибке)	Сообщение об ошибке
acsUrl	ANS..*	(при 3DS платеже)	URL для перехода на ACS
paReq	ANS..*	(при 3DS платеже)	Payment Authentication Request
termUrl	ANS..*	(при 3DS платеже)	URL для возврата с ACS

Коды ошибок (поле success):

Значение	Описание
0	Обработка запроса прошла без системных ошибок
2	Заказ с таким номером не найден
5	Доступ запрещён
5	Пользователь, осуществляющий вызов сервиса, должен изменить свой пароль
7	Системная ошибка

Пример запроса POST:

```
userName=username&mdOrder=65401edc-3fa1-4112-87fd-a569ca69fb6a&bindingId=41954212-70a 7-4eae-8430-90c1a87beda7&password=testPwd
```

Пример ответа:

```
{"info":"Ваш платёж обработан, происходит  
переадресация...","redirect":"finish.html?login=username&password=testPwd&orderId=654  
01edc-3fa1-4112-87fd-a569ca69fb6a","success":0}\
```

11.3.3 Запрос деактивации связи

Для того, чтобы сделать существующую связку неактивной, используется запрос unBindCard (см. раздел [Координаты подключения](#)).

Параметры запроса:

Название	Тип	Обязательно	Описание
userName	AN..30	да	Логин магазина, полученный при подключении.
password	AN..30	да	Пароль магазина, полученный при подключении.

bindingId	AN..255	да	Идентификатор связки созданной при оплате
			заказа или использованной для оплаты. Присутствует только если магазину разрешено создание связок.

Параметры ответа:

Название	Тип	Обязательно	Описание
errorCode	N3	нет	Код ошибки. Может отсутствовать если результат не привел к ошибке.
error	Boolean	Нет	Показывает наличие или отсутствие ошибки при обработке запроса
errorMessage	AN..512	нет	Описание ошибки

Коды ошибок (поле errorCode):

Значение	Описание
0	Обработка запроса прошла без системных ошибок
5	Доступ запрещён
5	Пользователь должен сменить свой пароль
2	Неверное состояние связки
2	Связка не найдена

Пример запроса POST:

```
userName=binding_api&password=testPwd&bindingId=fd3afc57-c6d0-4e08-aaef-1b7cfecb093dc
```

Пример ответа:

```
{"errorCode": "2", "errorMessage": "Binding isn't active"}
```

11.3.4 Запрос активации связки

Для активации деактивированной ранее связки используется запрос bindCard (см. раздел [Координаты подключения](#)).

Параметры запроса:

Название	Тип	Обязательно	Описание
userName	AN..30	да	Логин магазина, полученный при подключении.
password	AN..30	да	Пароль магазина, полученный при подключении.
bindingId	AN..255	да	Идентификатор связки созданной при оплате заказа или использованной для оплаты. Присутствует только если магазину разрешено создание связок.

Параметры ответа:

Название	Тип	Обязательно	Описание
errorCode	N3	нет	Код ошибки. Может отсутствовать если результат не привел к ошибке.

errorMessage	AN..512	При наличии ошибки	Описание ошибки
error	boolean	да	Показывает наличие ошибки при обработке запроса.
errorMessage	AN..512	нет	Описание ошибки.

Коды ошибок (поле errorCode):

Значение	Описание
0	Обработка запроса прошла без системных ошибок
5	Доступ запрещён
5	Пользователь должен сменить свой пароль
2	Неверное состояние связки
2	Связка не найдена

Пример запроса POST:

```
userName=binding_api&password=testPwd&bindingId=fd3afc57-c6d0-4e08-aaef-1b7cfec093dc
```

Пример ответа:

```
{"errorCode":"2","errorMessage":"Binding is active"}
```

11.3.5 Запрос списка возможных связок для мерчанта

Для получения списка связок по идентификатору клиента используется запрос getBindings (см. раздел [Координаты подключения](#)).

Параметры запроса:

Название	Тип	Обязательно	Описание
clientId	AN..255	да	Номер (идентификатор) клиента в системе магазина, переданный при регистрации заказа. Присутствует только если магазину разрешено создание связок.

Параметры ответа:

Название	Тип	Обязательно	Описание
errorCode	N1	да	Код завершения
errorMessage	ANS..*	(при ошибке)	Сообщение об ошибке
error	boolean	Да	Показывает наличие ошибки при обработке запроса
Элемент binding (состоит из bindingId, maskedPan, cardHolderName и expiryDate):			
bindingId	AN..255	нет	Идентификатор связки созданной при оплате заказа или использованной для оплаты. Присутствует только если магазину разрешено создание связок.

maskedPan	N..19	нет	Маскированный номер карты, которая использовалась для оплаты. Указан только
			после оплаты заказа.
expiryDate	N6	нет	Срок истечения действия карты в формате YYYYMM. Указан только после оплаты заказа.

Коды ошибок (поле errorCode):

Значение	Описание
0	Обработка запроса прошла без системных ошибок
1	[clientId] не задан
2	Информация не найдена
5	Доступ запрещен

Пример запроса POST:

```
userName=binding_api&password=testPwd&clientId=client
```

Пример ответа:

```
{"bindings": [{"bindingId": "fd3afc57-c6d0-4e08-aaef-1b7cfec093dc", "maskedPan": "4000 00** **** **02", "expiryDate": "201512"}], "errorCode": "0", "errorMessage": "Успешно"}
```

12 Рекуррентные платежи

Рекурренты проводятся с использованием связок в основе которых лежат 3D-Secure платежи

У магазина должны быть разрешения:

- 1) Can use Verified by Visa или Can use SecureCode
- 2) Merchant is allowed to initialize recurring transactions
- 3) Merchant is allowed to use bindings
- 4) SSL payment allowed

Возможно использование двух разных магазинов для первичных и вторичных рекуррентов с разрешениями 1-2 и 3-4 соответственно. Первый магазин работает строго по 3D-Secure без разрешения SSL платежей, второй использует связки.

Два рекуррентных платежа:

1) Первичный платеж: делаем связку типа TDS (с использованием 3D-Secure) одновременно с созданием первичного рекуррентного платежа, для этого регистрируем обычный платеж с указанием clientId и jsonParams содержащего recurringExpiry и recurringInitialize, пример запроса:

```
http://localhost:7001/payment/rest/register.do?orderNumber=20201224004&clientId=88&amount=100&currency=643&language=ru&password=probe&returnUrl=finish.html&userName=probe&jsonParams={"recurringExpiry":20220111,"recurringFrequency":14,"recurringInitialize":true}
```

После получения нового orderId проводим ЗДС платеж через обычную страницу ввода карт.

В случае успешной ЗДС авторизации будет создана связка типа TDS (полностью успешный 3D-Secure) и новый идентификатор рекуррентного платежа recurringId, его узнать запросом к getOrderStatus.do

Пример запроса:

```
http://localhost:7001/payment/rest/getOrderStatus.do?orderId=97a6682d-14ad-47db-a199-06a58bb296f7&password=probe&userName=probe
```

Пример ответа:

```
{"expiration":"201912","cardholderName":"jkhkjhkjkjhkjhkjh","depositAmount":100,"currency":"810","approvalCode":"123456","authCode":2,"clientId":"89","bindingId":"6e8ad2b4-64f8-4da4-bb44-67794ef8442e","params":{},"recurringId":"17d99f0b-2a26-41c7-8d50-24f7aa5c7bf8","ErrorCode":"0","ErrorMessage":"Успешно","OrderStatus":2,"OrderNumber":"20190211002","Pan":"456396**1999","Amount":100,"Ip":"127.0.0.1","SvfeResponse":"000"}
```

2) Вторичный рекуррентный платеж по связке типа TDS (в системе уже должны быть связка для clientId=88 полученная на первом этапе)

Платеж регистрируется с указанием recurringId от первого платежа в &jsonParams={"recurringId":"***"}

Пример запроса регистрации заказа вторичного рекуррента:

```
https://localhost:7001/payment/rest/register.do?orderNumber=20220419011&amount=123&clientId=88&currency=051&language=ru&password=probe_bind&returnUrl=finish.html&userName=probe_bind&jsonParams={"recurringId":"ad49eade-d0cd-496d-8e38-1b12eb580402"}
```

Магазины могут быть разные: первичный платеж был создан под магазином probe, вторичный под probe_bind, у вторичного магазина должны быть права на связки первичного

Проводится вторичный рекуррент запросом на /rest/paymentOrderRecurring.do с указанием идентификатора recurringId от первого платежа.

пример формы:

```
<form
action="http://localhost:7001/payment/rest/paymentOrderRecurring.do?userName=probe&password=probe"
method="POST">
    <input type="text" size="35" name="mdOrder" value="">
    <input type="text" name="recurringId" value="912dae1c-25ab-419d-8f78-499c088b2fb3">
    <input type="Submit" name="" value="PAY">
</form>
```

13 P2P платежи

Регистрация платежей для перевода с карты на карту делается аналогично регистрации обычных платежей, но в jsonParam надо добавить подпараметры "transaction_type":"transfer".

У магазина должно быть разрешение P2P transaction allowed

Пример запроса регистрации P2P платежа:

[https://localhost/payment/rest/register.do?orderNumber=20211108002&amount=125¤cy=643&language=ru&password=probe&returnUrl=finish.html&userName=probe&jsonParams={"transaction_type":"transfer"}](https://localhost/payment/rest/register.do?orderNumber=20211108002&amount=125¤cy=643&language=ru&password=probe&returnUrl=finish.html&userName=probe&jsonParams={)

Пример HTML формы ввода карт для p2p платежа

```
<form name="tranForm" action="/payment/rest/processformtransfer.do" method="POST">
    <input type="hidden" name="MDORDER">
    <input type="hidden" name="$PAN">
    <input type="hidden" name="$CVC">
    <input type="hidden" name="YYYY">
    <input type="hidden" name="MM">
    <input type="hidden" name="TEXT">
    <input type="hidden" name="P2P_PAN">
    <input type="hidden" name="P2P_EXPIRY">
</form>
```

Запрос с параметрами карт должен быть отправлен на адрес processformtransfer.do

Аналогично обычному платежу \$PAN, \$CVC, YYYY, MM – параметры дебетуемой карты.

MDORDER – уникальный идентификатор заказа полученный на этапе регистрации.

TEXT – имя держателя дебетуемой карты.

P2P_PAN, P2P_EXPIRY – параметры кредитуемой карты, где P2P_EXPIRY – срок действия кредитуемой карты, необязательный параметр, должен иметь форму YYYYMM

14 P2Pcredit платежи

Регистрация платежей для кредитования карты делается аналогично регистрации обычных платежей, но в параметр jsonParam надо добавить подпараметры "transaction_type":"p2p_credit".

У магазина должно быть разрешение P2P Credit (OCT) allowed

Пример запроса регистрации P2Pcredit платежа:

[https://localhost/payment/rest/register.do?orderNumber=20211108002&amount=125¤cy=643&language=ru&password=probe&returnUrl=finish.html&userName=probe&jsonParams={"transaction_type":"p2p_credit"}](https://localhost/payment/rest/register.do?orderNumber=20211108002&amount=125¤cy=643&language=ru&password=probe&returnUrl=finish.html&userName=probe&jsonParams={)

Пример HTML формы ввода карты для P2Pcredit платежа

```
<form name="tranForm" id="tranForm" action="/payment/rest/processformtransfer.do" method="POST">
    <input type="text" name="MDORDER">
    <input type="text" name="$PAN">
    <input type="text" name="YYYY">
    <input type="text" name="MM">
    <input type="text" name="TEXT">
</form>
```

Запрос с параметрами карт должен быть отправлен на адрес processformtransfer.do

\$PAN, YYYY, MM – параметры кредитуемой карты.

MDORDER – уникальный идентификатор заказа полученный на этапе регистрации.

TEXT – имя держателя кредитуемой карты.

15 Использование связок при P2P и P2P_credit

Если надо создать или использовать связку при p2p или p2p_credit, то надо добавить параметр clientId в регистрацию заказа. Тогда при успешной операции будет создана/обновлена связка.

Если связка уже есть, то при регистрации заказа тоже нужен clientId, а вместо ввода карты для p2p или p2p_credit нужно использовать rest/processBindingForm.do с bindingId, пример формы:

```
<form action="https://localhost/payment/rest/processBindingForm.do" method="POST">
    <input type="text" name="orderId" >
    <input type="text" name="bindingId">
    <input type="text" name="bindingId">
    <input type="text" name="cvc">
</form>
```

Если CVC2/CVV2 при платеже по связке не используется, то данное поле можно оставить пустым, но в параметр cvv обязательно должен присутствовать при вызове processBindingForm.do

Особенности операции P2P_credit по связке.

Если в качестве основы использовалась связка созданная по обычному платежу, то в качестве карты зачисления будет использована карта списания из первичного платежа

Если в качестве основы использовалась связка созданная по P2P платежу, то в качестве карты зачисления будет использована карта зачисления из первичного платежа.

16 3DS2 with card on client side

Регистрируете заказ запросом

<https://ipay.arca.am/payment/rest/register.do?orderNumber=20220930015&amount=1230¤cy=643&language=ru&password=probe&returnUrl=finish.html&userName=probe>

При связке карты или оплаты с использованием связки нужно добавить clientId в запрос регистрации. Получаете ответ на запрос регистрации заказа похожий на этот с идентификатором нового заказа в поле orderId, он же MDORDER или mdOrder в последующих запросах.

```
{"errorCode":0,"orderId":"70d7dcc5-c0d8-409a-a811-
```

```
50f5cc1988f2","formUrl":"http://localhost/ipay_3ds2/merchants/probe/payment_ru.html?mdOrder=70d7dcc5-c0d8-409a-a811-50f5cc1988f2","errorCodeString":"0","error":false}
```

Делаете POST запрос на <https://ipay.arca.am/payment/rest/threeds2/getUrls.do> с параметром запроса mdOrder=70d7dcc5-c0d8-409a-a811-50f5cc1988f2 - это новый идентификатор заказа полученный на шаге регистрации заказа

В отдельном потоке параллельно посыпаете данные карты, идентификатор запроса и другие параметры на адрес <https://ipay.arca.am/payment/rest/processform.do> POST запросом:

```
MDORDER=70d7dcc5-c0d8-409a-a81150f5cc1988f2
&EXPIRY=202401&PAN=4012000000003101&MM=01&YYYY=2024&TEXT=IvanovIvan&CVC=123
&language=ru&bindingNotNeeded=true
```

Сперва getUrl.do вернет ответ { "is3Ds2Eligible":false,"completed":false }, надо повторять запрос пока он не вернет подобный ответ с "is3Ds2Eligible":true, "completed":true

```
{"is3Ds2Eligible":true,"threeDSMethodURLServer":"https://localhost/api/v1/client/gather?threeDSServerTransID=d7f82c73-fb31-4506-9db0-e86d53ccabee","threeDSServerTransID":"d7f82c73-fb31-4506-9db0-e86d53ccabee","completed":true}
```

При присутствии дополнительных данных в **getUrl.do** могут вернуться следующие параметры:

- threeDSMethodURL
- threeDSMethodDataPacked

Пример:

```
{"is3Ds2Eligible":true, "threeDSServerTransID":"d5c902cb-aac1-4df1-9a5c-8f0f0fc46c8f","threeDSMethodURL":"https:// localhost /ma?param=aHR0cHM6Ly9hY3MyLmFyY2EuYW06OTc0NC9hY3MvYXBpLzNkcIvdGhyZWVEU01ldGhvZA==","threeDSMethodURLServer":"https://localhost/api/v1/client/gather?threeDSServerTransID=d5c902cb-aac1-4df1-9a5c-8f0f0fc46c8f","threeDSMethodDataPacked":"eyJ0aHJIZURTTWV0aG9kTm90aWZpY2F0aW9uVVJMIjoiaHR0cHM6Ly9pcGF5LmFyY2EuYW0vYXBpL3YxL2Fjcy9ub3RpZmljYXRpb24_dGhyZWVEU1NlcnZlclRyYW5zSUQ9ZDVjOTAyY2ItYWFjMS00ZGYxLTlhNWMtOGYwZjBmYzQ2YzhmlividGhyZWVEU1NlcnZlclRyYW5zSUQiOjknWM5MDJjYi1hYWMxLTRkZjEtOWE1Yy04ZjBmMGZjNDZjOGYifQ"}
```

Потом, если респонс processform.do еще не пришел, значит оно ждет 10 секунд, пока вы отправите POST запросом браузера пользователя на API:

<https://ipay.arca.am/api/v1/client>

с параметрами:

threeDSServerTransID=d7f82c73-fb31-4506-9db0-e86d53ccabee

&clientInfo={ параметры JSON которые описаны в отдельном документе 3DS2 - sv_3ds-server-1.32_dev-ref_en_core_20201014.pdf}

Пример:

```
{"userAgent":"PSPayClient","colorDepth":24,"screenHeight":1728,"screenWidth":1079,"javaEnabled":false,"browserLanguage":"en","browserTimeZoneOffset":-120,"browserAcceptHeader":"/","mobile":false}
```

При присутствии параметров **threeDSMethodURL** и **threeDSMethodDataPacked** сразу после отправки clientInfo необходимо в браузере клиента сформировать hidden iframe и в нем методом POST отправить на **threeDSMethodURL** значение **threeDSMethodDataPacked** в параметре threeDSMethodData.

Пример:

```
if (data['threeDSMethodURL'] != null && data['threeDSMethodURL'] !='null') {
    var iframe = document.createElement('iframe');
    iframe.name = 'acs_' + Date.now();
    iframe.src = '#';
    iframe.style.display = 'none';
    iframe = document.body.appendChild(iframe);
    var form = document.createElement('form');
    form.target = iframe.name;
    form.method = 'post';
    form.action = data['threeDSMethodURL'];
    var param = document.createElement('input');
    param.type = 'hidden';
    param.name = 'threeDSMethodData';
    param.value = data['threeDSMethodDataPacked'];
    param = form.appendChild(param);
    form = iframe.appendChild(form);
    form.submit();}
```

Если 3DS2 прошла, то запрос на processform.do выдаст набор данных для перенаправления кардхолдера на ACS банка эмитента.

acsUrl=https://acs2.arca.am/acs/api/3ds2/creqbrw

cReq=BASE64 закодированный Challenge Request

Ваше ПО должно отправить браузер кардхолдера на ACS подобной формой:

```
<form id="threeDS2" method="post" action="">
    <input type="hidden" id="CReq" name="creq"/>
</form>
```

Регистр параметров в форме важен!

После завершения аутентификации пользователя на ACS браузер пользователя вернется к вам на адрес returnUrl указанный в запросе регистрации заказа register.do.

ACS банка эмитента самостоятельно выполнит необходимые запросы в МПС для завершения платежа. Вам после возвращения браузера пользователя нужно будет проверить статус заказа с помощью запроса getOrderStatus.do.

Так последовательность этих запросов выглядит в браузере:

Первый запрос на getUrl.do



404 Not Found

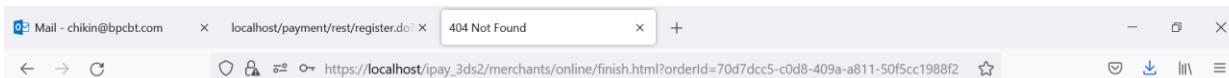
nginx/1.15.1

The screenshot shows the Network tab of a browser developer tools interface. It lists several requests made during the session:

- POST /getUrls.do**: Initiated by jquery-1.8.1..., Response status: 200, Size: 42... KB. Headers include Content-Type: application/json.
- POST /processform.do**: Initiated by jquery-1.8.1..., Response status: 200, Size: 21... KB. Headers include Content-Type: application/json.
- POST /getUrls.do**: Initiated by jquery-1.8.1..., Response status: 200, Size: 42... KB. Headers include is3Ds2Eligible: false, completed: false.
- GET /payment.html?mdOrder=70d7dcc5-c0d8-409a subdocument**: Initiated by NS_BINDING_ABORTED.
- POST /gather?threeDSserverTransID=d7f82c73-fb31~ subdocument**: Initiated by h..., Response status: 200, Size: 49.06 KB.
- POST /client**: Initiated by gather:145 (x...), Response status: 200, Size: 0.6 KB.
- GET /finish.html?orderId=70d7dcc5-c0d8-409a-a811**: Initiated by jquery.payme..., Response status: 404, Size: 369.6 KB.
- GET /favicon.ico**: Initiated by FaviconLoad..., Response status: 404, Size: 16... KB.

At the bottom, there is a summary: 9 запросов | 49,59 КБ / 52,49 КБ передано | Передано за: 4,21 с | DOMContentLoaded: 16 мс | load: 19 мс.

Отправка данных карты:



404 Not Found

nginx/1.15.1

The screenshot shows the Network tab of a browser developer tools interface. It lists several requests made during the session:

- POST /getUrls.do**: Initiated by jquery-1.8.1..., Response status: 200, Size: 42... KB. Headers include Content-Type: application/json.
- POST /processform.do**: Initiated by jquery-1.8.1..., Response status: 200, Size: 21... KB. Headers include Content-Type: application/json.
- POST /getUrls.do**: Initiated by jquery-1.8.1..., Response status: 200, Size: 42... KB. Headers include Content-Type: application/json.
- GET /payment.html?mdOrder=70d7dcc5-c0d8-409a subdocument**: Initiated by NS_BINDING_ABORTED.
- POST /gather?threeDSserverTransID=d7f82c73-fb31~ subdocument**: Initiated by h..., Response status: 200, Size: 49.06 KB.
- POST /client**: Initiated by gather:145 (x...), Response status: 200, Size: 0.6 KB.
- GET /finish.html?orderId=70d7dcc5-c0d8-409a-a811**: Initiated by jquery.payme..., Response status: 404, Size: 369.6 KB.
- GET /favicon.ico**: Initiated by FaviconLoad..., Response status: 404, Size: 16... KB.

In the middle section, there is a detailed view of the POST /processform.do request showing JSON parameters:

```

MDORDER: "70d7dcc5-c0d8-409a-a811-50f5cc1988f2"
$EXPIRY: "202401"
$PAN: "4012000000003101"
MM: "01"
YY: "2024"
TEXT: "VeraAdmin"
$CVC: "123"
language: "ru"
bindingNotNeeded: "true"
jsonParams: "0"

```

At the bottom, there is a summary: 9 запросов | 49,59 КБ / 52,49 КБ передано | Передано за: 4,21 с | DOMContentLoaded: 16 мс | load: 19 мс.

Получен успешный ответ на getUrls.do



404 Not Found

nginx/1.15.1

This screenshot shows the Network tab of a browser developer tools interface. It lists several requests made during the payment process:

- POST /getUrls.do (jquery-1.8.1...)
- POST /processform.do (jquery-1.8.1...)
- POST /getUrls.do (jquery-1.8.1...)
- POST /getUrls.do (jquery-1.8.1...)
- GET /payment.html?mdOrder=70d7dcc5-c0d8-409a subdocument (NS_BINDING_ABORTED)
- POST /gather?threeDServerTransID=d7f82c73-fb31- subdocument (h... 49,06 kB)
- POST /client (gather:145 x... h... 1,71 kB)
- GET /finish.html?orderId=70d7dcc5-c0d8-409a-a811 (jquery.payme... h... 369 kB)
- GET /favicon.ico (FaviconLoad... h... юцировано)

Details for the last request (GET /favicon.ico) include:

- Initiator: jquery-1.8.1...
- Type: pl...
- Size: 16...
- Response: NS_BINDING_ABORTED

Запрос для начала отправки данных браузера:

This screenshot shows the Network tab of a browser developer tools interface, focusing on the final step of the payment process:

- POST /gather?threeDServerTransID=d7f82c73-fb31- subdocument (h... 49,06 kB)
- POST /client (gather:145 x... h... 1,71 kB)
- GET /finish.html?orderId=70d7dcc5-c0d8-409a-a811 (jquery.payme... h... 369 kB)
- GET /favicon.ico (FaviconLoad... h... юцировано)

Details for the first POST request (POST /gather?threeDServerTransID=d7f82c73-fb31- subdocument) include:

- Initiator: jquery-1.8.1...
- Type: pl...
- Size: 48...
- Response: NS_BINDING_ABORTED

Details for the second POST request (POST /client) include:

- Initiator: jquery-1.8.1...
- Type: pl...
- Size: 42...
- Response: NS_BINDING_ABORTED

Details for the third GET request (GET /finish.html?orderId=70d7dcc5-c0d8-409a-a811) include:

- Initiator: jquery-1.8.1...
- Type: pl...
- Size: 22...
- Response: NS_BINDING_ABORTED

Ниже представлен ответ:

404 Not Found

nginx/1.15.1

Статус	Метод	Домен	Файл	Инициатор	Тип	Передано	Размер	Заголовки	Куки	Запрос	Ответ	Тайминги	Защита
200	POST	localhost	getUrls.do	jquery-1.8.1....	pl...	257 6	42...	HTML					
200	POST	localhost	processform.do	jquery-1.8.1....	pl...	429 6	21...						
200	POST	localhost	getUrls.do	jquery-1.8.1....	pl...	257 6	42...						
200	POST	localhost	getUrls.do	jquery-1.8.1....	pl...	441 6	22...						
400	GET	localhost	payment.html?mdOrder=70d7dcc5-c0d8-409a-...	subdocument	NS_BINDING_ABORTED								
200	POST	localhost	gather?threeDSServerTransID=d7fb82c73-fb31-	subdocument	h...	49,06 kB	48...						
200	POST	localhost	client	gather:145 (x...)	h...	1,71 kB	0 6						
404	GET	localhost	finish.html?orderId=70d7dcc5-c0d8-409a-a81...	jquery.payme...	h...	369 6	16...						
404	GET	localhost	favicon.ico	FaviconLoad...	h...	юшировано	16...						

```
MODULE: client_1590016(),
screenWidth: client.clientWidth(),
colorDepth: client.colorDepth,
screenHeight: screen && screen[1] ? screen[1] : 0,
screenWidth: screen && screen[0] ? screen[0] : 0,
plugins: client.getPlugins(),
javaEnabled: client.javaEnabled(),
browserLanguage: client.getLanguage(),
browserTimeZone: Intl.DateTimeFormat().resolvedOptions().timeZone,
browserTimezoneOffset: new Date().getTimezoneOffset(),
javascriptEnabled: true //always true here, because if JS were not enabled this would fail.
};

var dataForm = encodeForm({
    threeDSServerTransID: threeDSServerTransID,
    clientInfo: JSON.stringify(info, null, 2)
});

var request = new XMLHttpRequest();
request.open("POST", "https://localhost/api/v1/client");
request.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
request.send(dataForm);
}

</script>
</head>
<body>
</body>
</html>
```

9 запросов | 49,59 kB / 52,49 kB передано | Передано за: 4,21 с | DOMContentLoaded: 16 мс | load: 19 мс

Фактическая отправка данных на /api/v1/client:

The screenshot shows a browser window with the following details:

- Address bar: `localhost/payment/rest/register.do`
- Page title: 404 Not Found
- Content: "404 Not Found" (in large bold font)
- Header: "nginx/1.15.1"
- Toolbar buttons: Инспектор, Консоль, Отладчик, Сеть, Стили, Профайлер, Память, Хранилище, Поддержка доступности, Приложение.
- Network tab (Сеть):
 - Request list:
 - 200 POST /getUrls.do (jquery-1.8.1...)
 - 200 POST /processform.do (jquery-1.8.1...)
 - 200 POST /getUrls.do (jquery-1.8.1...)
 - 200 POST /getUrls.do (jquery-1.8.1...)
 - 200 GET /payment.html?mdOrder=70d7dcc5-c0d8-409a-a811-50f5cc1988f2 (NS_BINDING_ABORTED)
 - 200 POST /gather?threeDServerTransId=d7f82c73-fb31-4506-9db0-e96d53cabeef (gather:145...)
 - 200 POST /client (gather:145...)
 - 404 GET /finish.html?orderId=70d7dcc5-c0d8-409a-a811-50f5cc1988f2 (jquery.payme...)
 - 404 GET /favicon.ico (FaviconLoad...)
 - Details for selected request (gather:145...):
 - Заголовки: threeDServerTransId: "d7f82c73-fb31-4506-9db0-e96d53cabeef", clientInfo: "(Mozilla/5.0+(Windows+NT+10.0;+Win64;x64;+rv:105.0)+Gecko/20100101+Firefox/105.0\n\\n)+(os:+\"Windows\",\\n+(version:+\"10\",\\n+(mobile:+false,\\n+(screenPrint\n\\n+(+\"Current+Resolution:+1280x720,+Available+Resolution:+1280x680,+Color+Depth:+24,+Device+XDP:+undefined,+Device+YDP:+undefined,+colorDepth:+24,\\n+(screenHeight:+\"1720\",\\n+(screenWidth:+\"1280\n\\n+(+\"plugins\n\\n+(+\"PDF+Viewer,+Chrome+PDF+Viewer,+Chromium+PDF+Viewer,+Microsoft+Edge+PDF+Viewer,+WebKit+built-in+PDF\",\\n+(+\"javaEnabled\":"+false,\\n+(+\"browserLanguage\":"+\"ru\",\\n+(+\"browserTimezone\":"+\"Europe/Moscow\n\\n+(+\"browserTimezoneOffset\":"+-180,\\n+(+\"javascriptEnabled\":"+true\\n))
 - Куки: None
 - Запрос: None
 - Ответ: None
 - Тайминги: None
 - Стек вызовов: None
 - Защита: None

173DS2 with Binding on client side

Все адреса и пароли тестовые в этом документе. У вас они будут другие.

Регистрируете заказ запросом

<https://ipay.arca.am/payment/rest/register.do?orderNumber=20220930015&amount=1230¤cy=643&language=ru&password=probe&returnUrl=https://finish.html&userName=probe>

При связке карты или оплаты с использованием связки нужно добавить clientId в запрос регистрации. Получаете ответ на запрос регистрации заказа похожий на этот с идентификатором нового заказа в поле orderId, он же MDORDER или mdOrder в последующих запросах.

```
{ "errorCode":0,"orderId":"70d7dcc5-c0d8-409a-a811-50f5cc1988f2","formUrl":"http://ipay.arca.am/payment/merchants/probe/payment_ru.html?mdOrder=70d7dcc5-c0d8-409a-a811-50f5cc1988f2","errorCodeString":"0","error":false}
```

Делаете POST запрос на <https://ipay.arca.am/payment/rest/threeds2/getUrls.do> с параметром запроса

mdOrder=70d7dcc5-c0d8-409a-a811-50f5cc1988f2 - это новый идентификатор заказа полученный на шаге регистрации заказа

В отдельном потоке параллельно посылаете данные байндинга, идентификатор запроса и другие параметры на адрес <https://ipay.arca.am/payment/rest/paymentOrderBinding.do> POST запросом:

Сперва getUrl.do вернет ответ {"is3Ds2Eligible":false,"completed":false}, надо повторять запрос пока он не вернет подобный ответ с "is3Ds2Eligible":true, "completed":true

```
{"is3Ds2Eligible":true,"threeDSMethodURLServer":"https://localhost/api/v1/client/gather?threeDServerTransID=d7f82c73-fb31-4506-9db0-e86d53ccabee","threeDServerTransID":"d7f82c73-fb31-4506-9db0-e86d53ccabee","completed":true}
```

При присутствии дополнительных данных в **getUrl.do** могут вернуться следующие параметры:

- threeDSMethodURL
- threeDSMethodDataPacked

Пример:

```
{"is3Ds2Eligible":true,"threeDServerTransID":"d5c902cb-aac1-4df1-9a5c-8f0f0fc46c8f","threeDSMethodURL":"https://localhost/ma?param=aHR0cHM6Ly9hY3MyLmFyY2EuYW06OTc0NC9hY3MvYXBpLzNkczIvdGhyZWVEU01ldGhvZA==","threeDSMethodURLServer":"https://localhost/api/v1/client/gather?threeDServerTransID=d5c902cb-aac1-4df1-9a5c-8f0f0fc46c8f","threeDSMethodDataPacked":"eyJ0aHJlZURTTWV0aG9kTm90aWZpY2F0aW9uVVJMIjoiaHR0cHM6Ly9pcGF5LmFyY2EuYW0vYXBpL3YxL2Fjcy9ub3RpZmljYXRpb24_dGhyZWVEU1NlcnZlclRyYW5zSUQ9ZDVjOTAyY2ItYWFjMS00ZGYxLTlhNWMtOGYwZjBmYzQ2YzhmiwidGhyZWVEU1NlcnZlclRyYW5zSUQiOjKjNWM5MDJjYi1hYWMxLTRkZjEtOWE1Yy04ZjBmMGZjNDZjOGYifQ"}
```

Потом, если респонс paymentOrderBinding.do еще не пришел, значит оно ждет 10 секунд, пока вы отправите POST запросом данные браузера пользователя на API:

<https://ipay.arca.am/api/v1/client>

с параметрами:

threeDServerTransID=d7f82c73-fb31-4506-9db0-e86d53ccabee

&clientInfo={ параметры JSON которые описаны в отдельном документе 3DS2 - sv_3ds-server-1.32_dev-ref_en_core_20201014.pdf }

Пример:

```
{"userAgent":"PSPayClient","colorDepth":24,"screenHeight":1728,"screenWidth":1079,"javaEnabled":false,
```

```
{"browserLanguage":"en","browserTimeZoneOffset":-120,"browserAcceptHeader":"/*/*","mobile":false}
```

При присутствии параметров **threeDSMethodURL** и **threeDSMethodDataPacked** сразу после отправки **clientInfo** необходимо в браузере клиента сформировать hidden iframe и в нем методом POST отправить на **threeDSMethodURL** значение **threeDSMethodDataPacked** в параметре **threeDSMethodData**.

Пример:

```
if (data['threeDSMethodURL'] != null && data['threeDSMethodURL'] !='null') {
    var iframe = document.createElement('iframe');
    iframe.name = 'acs_'+ Date.now();
    iframe.src = '#';
    iframe.style.display = 'none';
    iframe = document.body.appendChild(iframe);
    var form = document.createElement('form');
    form.target = iframe.name;
    form.method = 'post';
    form.action = data['threeDSMethodURL'];
    var param = document.createElement('input');
    param.type = 'hidden';
    param.name = 'threeDSMethodData';
    param.value = data['threeDSMethodDataPacked'];
    param = form.appendChild(param);
    form = iframe.appendChild(form);
    form.submit();}
```

Если 3DS2 прошла, то запрос на **paymentOrderBinding.do** выдаст набор данных для перенаправления кардхолдера на ACS банка эмитента.

Пример:

```
{"userAgent":"PSPPayClient","colorDepth":"24","screenHeight":1728,"screenWidth":1079,"javaEnabled":false,
"browserLanguage":"en","browserTimeZoneOffset":-120,"browserAcceptHeader":"/*/*","mobile":false}
acsUrl=https://acs2.arca.am/acs/api/3ds2/creqbrw
```

cReq=BASE64 закодированный Challenge Request

Ваше ПО должно отправить браузер кардхолдера на ACS подобной формой:

```
<form id="threeDS2" method="post" action=""
<input type="hidden" id="CReq" name="creq"/>
</form>
```

Регистр параметров в форме важен!

После завершения аутентификации пользователя на ACS браузер пользователя вернется к вам на адрес **returnUrl** указанный в запросе регистрации заказа **register.do**.

ACS банка эмитента самостоятельно выполнит необходимые запросы в МПС для завершения платежа. Вам после возвращения браузера пользователя нужно будет проверить статус заказа с помощью запроса **getOrderStatus.do**.

Важно: processform.do в картинках это тоже самое что и paymentOrderBinding.do

Последовательность этих запросов выглядит в браузере так же как и в картинках пункта 16:

18 Приложения

18.1 Приложение №1. Список значений action code.

Action code	error_id	error_message	Описание
-20010	-20010	BLOCKED_BY_LIMIT	Транзакция отклонена по причине того, что размер платежа превысил установленные лимиты Банком-эмитентом
-9000	-9000	Started	Состояние начала транзакции
-2021	-2021	TDS2_UNKNOWN_STATUS_IN_RREQ TDS2 unknown status in RReq	Получен неизвестный статус транзакции в RReq
-2019	-2019	Decline by iReq in PARes	PARes от эмитента содержит iReq, вследствие чего платеж был отклонен
-2018	-2018	Declined. DS connection timeout	Directory server Visa или MasterCard либо недоступен, либо в ответ на запрос вовлеченности карты (VeReq) пришла ошибка связи. Это ошибка взаимодействия платежного шлюза и серверов МПС по причине технических неполадок на стороне последних.
-2016	-2016	Declined. VeRes status is unknown	Это означает, что банк-эммитент не готов (в данный момент времени) провести 3ds транзакцию (например, не работает ACS банка). Мы не можем определить вовлечена ли карта в 3d secure
-2015	-2015	Decline by iReq in VERes	VERes от DS содержит iReq, вследствие чего платеж был отклонен
-2014	-2014	TDS2_CHALLENGE_CANCELED TDS2 Challenge canceled	Кардхолдер нажал кнопку Cancel во время ЗДС2 аутентификации
-2013	-2013	Исчерпаны попытки оплаты	Исчерпаны попытки оплаты
-2012	-2012	Operation not supported	Данная операция не поддерживается
-2011	-2011	Declined. PaRes status is unknown	Карта определена как вовлеченная в карта 3d secure, но банк-эммитент не готов (в данный момент времени) провести 3ds транзакцию
-2009	-2009	TDS2_DECLINED_BY_ARES TDS2 declined by ARes	TDS2 отклонено по статусу транзакции полученного Ares
-2007	2007	Decline. Payment time limit	Истек срок, отведенный на ввод данных карты с момента регистрации платежа (в данный момент таймаут наступит через 20 минут)
-2006	2006	Decline. 3DSec decline	Означает, что эмитент отклонил аутентификацию.

-2005	-2005	Decline. 3DSec sign error	Означает, что мы не смогли проверить подпись эмитента, то есть PARes был читаемый, но подписан неверно.
-2004	-2004	TDS2_CHALLENGE_FAILED TDS2 Challenge failed	Кархолдер не прошел ЗДС2 аутентификацию
-2002	2002	Decline. Payment over limit	Транзакция отклонена по причине того, что размер платежа превысил установленные лимиты. Примечание: имеется в виду либо лимиты Банка-эквайера на дневной оборот Магазина, либо лимиты Магазина на оборот по одной карте, либо лимит Магазина по одной операции)
-2001	2001	Decline. IP blacklisted	Транзакция отклонена по причине того, что IP-адрес Клиента внесен в черный список.
-2000	2000	Decline. PAN blacklisted	Транзакция отклонена по причине того, что карта внесена в черный список
-100	-100	no_payments_yet	Не было попыток оплаты.
-1	-1	sv_unavailable	Истекло время ожидания ответа от процессинговой системы.
0	0	Approved.	Платеж успешно прошел
1	1	Declined. Honor with id	Для успешного завершения транзакции, требуется подтверждение личности. В случае интернет-транзакции (соот-но и в нашем) невозможно, поэтому считается как declined.
5	5	Decline. Unable to process	Отказ сети проводить транзакцию.
100	100	Decline. Card declined	Банкээмитент запретил интернет транзакции по карте.
101	101	Decline. Expired card	Истек срок действия карты.
103	103	Decline. Call issuer	Нет связи с Банком-Эмитентом. Торговой точке необходимо связаться с банком-эмитентом.
104	104	Decline. Card declined	Попытка выполнения операции по счету, на использование которого наложены ограничения.
107	107	Decline. Call issuer	Следует обратиться к Банку-Эмитенту.
109	109	Decline. Invalid merchant	Неверно указан идентификатор мерчанта/терминала (для завершения и предавторизация с разными идентификаторами)
110	110	Decline. Invalid amount	Неверно указана сумма транзакции
111	111	Decline. No card record	Неверный номер карты
116	116	Decline. Decline. Not enough money	Сумма транзакции превышает доступный остаток средств на выбранном счете.
120	120	Decline. Not allowed	Отказ в проведении операции - транзакция не разрешена эмитентом. Код ответа платежной сети - 57. Причины отказа необходимо уточнять у эмитента.

121	121	Decline. Excds wdrwl limit	Предпринята попытка выполнить транзакцию на сумму, превышающую дневной лимит, заданный банком-эмитентом
123	123	Decline. Excds wdrwl ltmt	Превышен лимит начисло транзакций: клиент выполнил максимально разрешенное число транзакций в течение лимитного цикла и пытается провести еще одну.
125	125	Decline. Card declined	Неверный номер карты. Попытка возврата на сумму, больше холда, попытка возврата нулевой суммы.
208	208	Decline. Card is lost	Карта утеряна
209	209	Decline. Card limitations exceeded	Превышены ограничения по карте
902	902	Decline. Invalid trans	Владелец карты пытается выполнить транзакцию, которая для него не разрешена.
903	903	Decline. Re-enter trans.	Предпринята попытка выполнить транзакцию на сумму, превышающую лимит, заданный банком-эмитентом
904	904	Decline. Format error	Ошибочный формат сообщения с точки зрения банка эмитента.
907	907	Decline. Host not avail.	Нет связи с Банком, выпустившим Вашу карту. Для данного номера карты не разрешена авторизация в режиме stand-in (этот режим означает, что эмитент не может связаться с платежной сетью и поэтому транзакция возможна либо в оффлайне с последующей выгрузкой в бэк офис, либо она будет отклонена)
909	909	Decline. Call issuer	Ошибка функционирования системы, имеющая общий характер. Фиксируется платежной сетью или банком-эмитентом.
910	910	Decline. Host not avail.	Банк-эмитент недоступен.
913	913	Decline. Invalid trans	Неправильный формат транзакции с точки зрения сети.
914	914	Decline. Orig trans not found	Не найдена транзакция (когда посыпается завершение или reversal или refund)
999	999	Declined by fraud	Отсутствует начало авторизации транзакции. Отклонено по фроду или ошибка 3dsec.
1001	1001	Decline. Data input timeout	Выставляется в момент регистрации транзакции, т.е. когда еще по транзакции не было введено данных карты
2001	200	Decline. Fraud	Фродовая транзакция (по мнению процессинга или платежной сети)
2003	2003	Decline. SSL restricted	SSL (He 3d-Secure/SecureCode) транзакции запрещены Магазину
2004	2004	SSL_WITHOUT_CVC_FOR BIDDEN SSL without CVC forbidden	Запрещен SSL платеж без ввода CVC

2005	2005	3DS rule failed	Платеж не соответствует условиям правила проверки по 3ds
8204	8204	Decline. Duplicate order	Такой заказ уже регистрировали (проверка по ordernumber)
9001	9001	RBS internal error	Внутренний код отказа РБС
71015	1015	Decline. Input error	Введены неправильные параметры карты
151017	1017	Decline. 3DSec comm error	3-D Secure - ошибка связи
151018	1018	Decline. Processing timeout	Таймаут в процессинге. Не удалось отправить
151019	1019	Decline. Processing timeout	Таймаут в процессинге. Удалось отправить, но не получен ответ от банка
341014	1014	Decline. General Error	Код отказа РБС
341041	1041	Decline. Refund failed	Ошибка возврата денежных средств