Банк АВАНГАРД

Техническая документация к сервису «Интернет-эквайринг»
Версия 4.1

Москва

26 августа 2021 г.

Оглавление

Описание изменений

Версия	Описание
4.1	Добавлен механизм оплаты по QR
4.0	Добавлен механизм оплаты заказа без повторного ввода карточных данных (раздел 1.1.3).
3.1	Добавляем сервис запрета оплаты (2.7). Клиент, вызывая данный API, запрещает нам принимать оплату по ticket
2.24	Добавлено поле “trn” в ответ get_order_info. Доступно при указании в теле запроса <version>3</version>
2.22	Добавлены рекомендации по безопасности (раздел 3.2) в случае использования сервиса уведомлений магазина о завершении процесса оплаты.
2.21	Убрана версия ответа из сервиса получения информации о заказе (2.3). В сервисы получения списка операций по номеру заказа и дате (2.5, 2.6) добавлены поля суммы возврата и суммы комиссии Банка. Добавлен код ответа сервисов 10 (некорректный формат суммы). Добавлены рекомендации по реализации взаимодействия с сервисами host2host в разделе 2.1.
2.2	Изменен формат ответа сервиса получения списка операций по номеру заказа (добавлено поле order_number). Добавлен сервис получения списка операций за определенную дату.
2.1	Добавлен отдельный статус для возвратов и частичных возвратов. В сервис получения списка операций по заказу добавлена расширенная информация о переходах по статусам внутри одного тикета.
2.0	Добавлен альтернативный вариант отправки заказа посредством HTML-формы (без использования протокола host2host). Добавлен сервис уведомления магазина по завершении процесса оплаты клиентом на стороне Интернет-эквайринга. Добавлена возможность частичного возврата заказа. Добавлены SOAP веб-сервисы в качестве альтернативы host2host-протоколу.

	Расширен набор допустимых статусов ответов системы. В сервис get_order_info добавлена расширенная информация о заказе. Добавлен сервис получения информации обо всех попытках оплаты заказа по его номеру.

1. Общая схема работы

1.1. Формирование заказа в системе Интернет-эквайринга

Заказ может быть сформирован в системе Интернет-эквайринга Банка Авангард двумя способами. Первый – по протоколу host2host (данные о заказе передаются в формате XML), второй – посредством отправки POST-запроса со всеми необходимыми параметрами на определенный адрес сервиса. Второй способ является более простым с точки зрения технической реализации, но часть данных передается в открытом для клиента виде.

1.1.1. Формирование заказа по протоколу host2host

После формирования заказа клиентом на стороне Вашего Интернет-магазина, перед непосредственно переходом к оплате, необходимо осуществить следующую последовательность действий:

1) Выполнить запрос к сервису регистрации заказа на стороне Интернет-эквайринга. В данном запросе передать все необходимые (см. подробное описание сервиса, **2.2**) детали заказа. В ответ сервис вернет уникальный идентификационный номер – ticket, который позволяет осуществить единичную попытку оплаты. Помимо этого ответ содержит коды успешной и неуспешной авторизации (соответственно, ok_code и failure_code), которые впоследствии могут быть переданы, если клиент после завершения процесса оплаты вернется по ссылке в Ваш Интернет-магазин. Полученные от сервиса регистрации параметры необходимо сохранить на своей стороне до полного завершения процесса оплаты.

2) Перенаправить клиента на адрес https://pay.avangard.ru/iacq/pay, передав в качестве параметра (с именем ticket) полученный от сервиса регистрации уникальный идентификационный номер (ticket). Передача параметра возможна как методом GET, так и методом POST.

   Пример корректного адреса (параметр передается методом GET): https://pay.avangard.ru/iacq/pay?ticket=1234567890ABCDEABCDE12345678901234567890

После перехода на сторону Интернет-эквайринга клиент заполняет информацию о своем платежном средстве и вводит (при необходимости) коды подтверждения. После совершения оплаты клиент возвращается по указанной при регистрации заказа (см. шаг 1) ссылке в Ваш Интернет-магазин. При этом GET-параметром result_code передается либо успешный код авторизации (ok_code) либо код ошибки (failure_code). Следует отметить, что учитывая специфику передачи данного параметра в открытом виде, не стоит полагаться на данный атрибут как полноценный признак успешности оплаты.

Настоятельно рекомендуется использовать его только для инициации запроса host2host на сервис получения информации о заказе. Помимо всего прочего, result_code может быть не передан вовсе, так как возврат в Интернет-магазин по заданной ссылке является сугубо прерогативой клиента и, соответственно, передача параметра ничем не гарантируется.

Получение актуального статуса заказа возможно как самостоятельно, так и посредством сервиса уведомлений об успешных оплатах. В первом случае после перехода клиента непосредственно к оплате на страницу Интернет-эквайринга, рекомендуется осуществлять с определенной периодичностью (например, раз в несколько секунд) запрос к сервису получения информации о заказе. Соответственно, после изменения статуса

заказа на «Исполнен», необходимо блокировать возможность перехода к оплате со стороны Вашего Интернет-магазина. Во избежание повторной оплаты клиентом того же самого заказа, перед каждым новым обращением к сервису регистрации рекомендуется проверять статусы предыдущих попыток оплаты. Возможен также вариант получения актуального статуса непосредственно от сервиса уведомлений, подробное его описание находится в разделе **3**.

Резюмируя вышесказанное, следует заострить внимание на ключевых моментах процесса оплаты по протоколу host2host:

1) Каждому заказу в Интернет-магазине может соответствовать как одна, так и несколько уникальных сессий оплаты, каждая из которых имеет свой ticket.
2) Перед перенаправлением клиента крайне желательно сначала убедиться, что для данного заказа ни одна из ранее начатых сессий оплаты не была завершена с успешным статусом. Если таких сессий нет, то посредством обращения к сервису регистрации необходимо получить новый уникальный идентификационный номер сессии (ticket) и перенаправить с ним в качестве параметра клиента на страницу Интернет-эквайринга.
3) Актуальный статус заказа можно получать самостоятельно либо от соответствующего сервиса после совершения успешной оплаты клиентом. Необходимо проверять статусы всех клиентских сессий оплаты, привязанных к данному заказу. Как только одна из них переходит в успешный статус, нужно блокировать возможность получения клиентом новых ticket'ов для этого заказа. Дополнительно рекомендуется ограничивать максимальное время опроса сервера Интернет-эквайринга на предмет изменения статуса заказа (например, одним часом).
4) Помимо проверки статуса по заданному интервалу времени рекомендуется осуществлять запрос на сервис получения информации о заказе после возврата клиента в Интернет-магазин с успешным статусом в качестве GET-параметра result_code.

1.1.2. Формирование заказа посредством POST-запроса

На стороне Вашего Интернет-магазина для перехода к оплате необходимо сформировать HTML-форму, передающую основные параметры о заказе (http-методом **POST** в кодировке **UTF-8**). Адрес сервиса обработки подобных запросов - https://pay.avangard.ru/iacq/post . На основании номера заказа система Интернет-эквайринга определит, осуществлялись ли успешные оплаты данного заказа. Если такие заказы не будут найдены, система автоматически перенаправит клиента на страницу оплаты. В противном случае клиенту будет предложено вернуться в Интернет-магазин. Список необходимых параметров указан в таблице 1.1. Описание типов данных, которые могут встречаться в описании сервисов, приведено в приложении 1.

В силу того, что запрос передается в фактически открытом для клиента виде и потенциально может быть скомпрометирован, в систему Интернет-эквайринга необходимо передать контрольную сумму заказа. В случае если контрольная сумма заказа некорректна, клиент будет перенаправлен на страницу ошибки. По желанию Интернет-магазина проверка контрольной суммы может быть отключена, но делать это настоятельно не рекомендуется.

Алгоритм вычисления контрольной суммы заказа:

signature = UPPER(MD5(UPPER(MD5(***shop_sign***) + MD5(***shop_id*** + ***order_number*** + ***amount***))))

, где

signature – непосредственно контрольная сумма, которая передается в качестве одного из параметров POST-запроса;

shop_sign – "подпись" Интернет-магазина (выдается техподдержкой);

shop_id – идентификатор магазина;

order_number – номер заказа;

amount – сумма заказа (в копейках);

MD5 – шестнадцатеричное представление хеша, сгенерированного по алгоритму MD5;

UPPER – перевод символов в верхний регистр;

Знак «+» (плюс) - конкатенация строк.

Подпись Интернет-магазина (shop_sign) не должна предаваться огласке третьим лицам, поэтому контрольная сумма обязательно должна вычисляться на стороне сервера (не на клиентской части).

Алгоритм последующей обработки заказа не отличается по своей сути от описанного в разделе **1.1.1**. Клиент заполняет информацию о платежном средстве, в случае успешной оплаты возвращается в Интернет-магазин. Единственным отличием является то, что Интернет-магазин может не располагать информацией о тикете, успешных и неуспешных кодах оплаты (ok_code/failure_code), поэтому в качестве дополнительных параметров введены ***back_url_ok*** и ***back_url_fail***. При успешной оплате клиент будет перенаправлен на первый из них, в противном случае – на второй. Если один из этих адресов не задан, клиент будет перенаправлен на адрес из параметра ***back_url***. При этом коды ok_code/failure_code по-прежнему передаются.

Получение актуального статуса заказа возможно как самостоятельно, так и посредством сервиса уведомлений об успешных оплатах (процесс полностью идентичен описанному в разделе **1.1.1**).

Пошагово процесс формирования заказа посредством POST-запроса выглядит так:

1)Сформировать HTML-форму со всеми необходимыми параметрами заказа и контрольной суммой (опционально). Контроль уникальности оплат осуществляется на стороне сервера Интернет-эквайринга по номеру заказа. При этом потенциально несколько оплат по одному заказу можно совершить, если одновременно со стороны Интернет-магазина будет возможность сформировать заказ по протоколу host2host и методом POST.

2)Настроить на стороне Интернет-магазина страницы, куда будет перенаправлен пользователь в случае успешной оплаты (***back_url_ok***), а также при различных ошибках либо досрочном возврате в Интернет-магазин (***back_url_fail***).

3)Получить актуальный статус заказа. Для этого либо нужно дождаться уведомления от системы Интернет-эквайринга (см. раздел **3**), либо самостоятельно получать расширенную информацию с определенной периодичностью. Допустимые статусы заказов в системе приведены в приложении 1.

1.1.3. Формирование заказа с использованием номера транзакции TXN

При формировании заказа можно передать связку параметров **card_num** (маскированный номер карты) и **txn** (номер ранее исполненной транзакции), чтобы пропустить 1-й шаг оплаты заказа с вводом карточных данных.

Номер TXN автоматически присваивается каждому заказу и его можно получить через запрос GetOrderInfo версии 4.

Оба параметра card_num и txn связаны между собой и при их наличии сработают дополнительные проверки.

Ошибка с кодом 120 «Некорректные данные для повторной оплаты» вернется в случае, если передан txn, но не передан card_num.

Ошибка с кодом 121 «Повторная оплата невозможна» вернется в случае, если по переданному txn не найден ранее оплаченный заказ или card_num не соответствует номеру карты найденного заказа.

1.1.4. Формирование заказа с использованием номера карты

При формировании заказа можно передать связку параметров **card_num** (полный номер карты) и exp_year ,exp_month, cvv, чтобы пропустить 1-й шаг оплаты заказа с вводом карточных данных. Такой режим работы по умолчанию не включен и включается отдельно для каждого магазина.

1.1.5. Получение изображения Qr-кода в ответе на запрос регистрации заказа

Для получения Изображения Qr-кода при формировании заказа нужно указать параметр return_qr_image = 1. В ответе на такой запрос будет присутствовать поле qr_png_base64. Пример как отобразить в html

Таблица 1.1. Параметры для формирования заказа посредством POST-запроса.

Имя параметра	Тип	Обязательность для заполнения	Описание параметра
shop_id	Number(10)	+	Уникальный идентификационный номер торговой точки
signature	Char(32)	+	Подпись заказа
amount	Number(24)	+	Сумма заказа, указывается в копейках РФ
order_number	Varchar(100)	+	Уникальный идентификационный номер заказа в системе Вашего Интернет-магазина. Предпочтительно использовать только цифры и символы латинского алфавита
order_description	Varchar(500)	+	Описание заказа

language	Varchar(2)	+	Язык заказа
back_url	Varchar(500)	+	URL ссылки перехода обратно в магазин со страницы оплаты
back_url_ok	Varchar(500)	-	URL ссылки перехода обратно в магазин в случае успешной оплаты
back_url_fail	Varchar(500)	-	URL ссылки перехода обратно в магазин в случае ошибок либо возврата без оплаты
client_name	Varchar(254)	-	Полное имя клиента
client_address	Varchar(254)	-	Полный адрес клиента
client_phone	Varchar(30)	-	Телефон клиента
client_email	Varchar(60)	-	Электронный почтовый адрес клиента
card_num	Varchar(16)	-	Маскированный номер карты вида 123456XXXXXX1234 для заказов с TXN Вместо X может быть любой символ. Полный номер карты для заказов с номером карты (раздел 1.1.4)
txn	Varchar(36)	-	Номер транзакции (раздел 1.1.3)
exp_year	Varchar(2)	-	Год экспирации карты (раздел 1.1.4)
exp_month	Varchar(2)	-	Месяц экспирации карты (раздел 1.1.4)
cvv	Varchar(5)	-	Cvv/cvc код (раздел 1.1.4)
is_qr	Number(1)	-	Признак оплаты по qr, 1- оплата по QR-коду, 0- оплата по карте (по умолчанию 0)

Пример HTML-формы запроса:

```
<form action="https://pay.avangard.ru/iacq/post" method="POST" accept-charset="UTF-8">
 <input type="hidden" name="shop_id" value="123456789"/>
 <input type="hidden" name="amount" value="30000"/>
 <input type="hidden" name="back_url" value="https://pay.avangard.ru/"/>
 <input type="hidden" name="back_url_ok" value="https://payavangard.ru/thank_you"/>
 <input type="hidden" name="back_url_fail" value="https://pay.avangard.ru/order"/>
 <input type="hidden" name="order_number" value="1234"/>
 <input type="hidden" name="order_description" value="Описание заказа"/>
 <input type="hidden" name="language" value="RU"/>
 <input type="hidden" name="client_name" value="Иванов Иван Иванович"/>
 <input type="hidden" name="client_phone" value="+74950000000"/>
 <input type="hidden" name="client_email" value="iacq@avangard.ru"/>
 <input type="hidden" name="client_address" value="г. Москва, ул. Садовническая 12"/>
 <input type="hidden" name="signature" value="5C426D69AA7AC904A9B7C94B42AEB311"/>
 <input type="hidden" name="language" value="RU"/>
 <input type="submit" value="Оплатить"/>
</form>
```

2. Сервисы host2host

2.1. Общая информация о сервисах

Обмен данными между серверами Интернет-магазина и Интернет-эквайрингом может осуществляться посредством как в формате XML, так и SOAP. Параметры в обоих случаях используются абсолютно идентичные. Подробное описание формата сообщений XML можно найти в разделе соответствующего сервиса, а WSDL описание для протокола SOAP находится по адресу:

https://pay.avangard.ru/iacq/AcqService?wsdl

URL-адреса сервисов для формата XML указаны в технической документации, для формата SOAP – в WSDL-описании.

Интернет-магазин может использовать любой удобный из предложенных форматов обращения. Для большей безопасности есть возможность ограничения host2host-запросов только с конкретного IP-адреса (включается через техподдержку).

Для обмена данных в формате XML допускается использовать кодировки **windows-1251** и **UTF-8,** система эквайринга будет возвращать ответ в кодировке запроса. Кодировка запроса берется из объявления xml (<?xml encoding=”…”?>). Кодировки http-запроса и данных внутри XML-сообщения обязательно должны совпадать. Для протокола SOAP допускается использовать **только** кодировку **UTF-8**.

Важным нюансом для обмена данными в формате XML является обязательное требование экранирования специальных символов в теле документа.

Для всех сервисов реализован механизм версионности для сохранения обратной совместимости при расширении предоставляемой информации. В настоящий момент поддерживаются версии с **1** по **4** (передаются в качестве опциональных параметров). Отличия между версиями можно найти в разделе описания конкретного сервиса.

Важная информация! В случае самостоятельного обращения к host2host-сервисам получения информации о заказе (см. раздел **2.3**) с целью контроля завершения процесса оплаты на стороне Интернет-эквайринга, **настоятельно рекомендуется** ограничить максимальный период запроса (например, одним часом), а также прекращать опрос сервисов в случае перехода заказа в конечный статус (отбракован, исполнен, возврат).

2.2. Регистрация заказа в системе Интернет-эквайринга

URL: https://pay.avangard.ru/iacq/h2h/reg

Сервис предназначен для регистрации информации о заказе в системе эквайринга. В качестве входных параметров передаются полные сведения о заказе, в ответ сервис возвращает ticket и служебную информацию – код ответа и детальное сообщение ответа. Ticket – это уникальный идентификационный номер заказа в системе, предназначенный для единственной попытки оплаты. В случае каких-либо ошибок в процессе оплаты клиентом повторное использование данного ticket'а невозможно.

Таблица 2.2.1. Входные параметры для сервиса регистрации заказа.

Имя параметра	Тип	Обязательность для заполнения	Описание параметра
shop_id	Number(10)	+	Уникальный идентификационный номер торговой точки
shop_passwd	Varchar(32)	+	Пароль для данной торговой точки
amount	Number(24)	+	Сумма заказа, указывается в копейках РФ
order_number	Varchar(100)	+	Уникальный идентификационный номер заказа в системе Вашего Интернет-магазина
order_description	Varchar(500)	+	Описание заказа
language	Varchar(2)	+	Язык заказа
back_url	Varchar(500)	+	URL ссылки перехода обратно в магазин со страницы оплаты
back_url_ok	Varchar(500)	-	URL ссылки перехода обратно в магазин в случае успешной оплаты
back_url_fail	Varchar(500)	-	URL ссылки перехода обратно в магазин в случае ошибок либо возврата без оплаты
client_name	Varchar(254)	-	Полное имя клиента*
client_address	Varchar(254)	-	Полный адрес клиента*
client_phone	Varchar(30)	-	Телефон клиента*
client_email	Varchar(60)	-	Электронный почтовый адрес клиента*

client_ip	Varchar(100)	-	IP-адрес клиента*, с которым он осуществил заказ
version	Number(1)	-	Версия формата ответа, для данного сервиса не поддерживается
card_num	Varchar(16)	-	Маскированный номер карты вида 123456XXXXXX1234 Вместо X может быть любой символ.
txn	Varchar(36)	-	Номер транзакции
is_qr	Number(1)	-	Признак оплаты по qr, 1- оплата по QR-коду, 0- оплата по карте (по умолчанию 0)
qr_ttl	Number	-	Время жизни qr кода в минутах
return_qr_image	Number(1)	-	Признак возврата изображения Qr-кода в ответе, 1- возвращать изображение, 0- не возвращать изображение (по умолчанию 0)

** - под клиентом подразумевается лицо, выступающее в роли покупателя в Вашем Интернет-магазине*

Входные параметры передаются в формате XML, передача возможна как методом GET, так и POST (предпочтительно), наименование параметра с входящим XML-сообщением – “xml” (без кавычек). Регистр элементов во входящем сообщении и их порядок следования не важен.

Пример входного сообщения:

```
<?xml version="1.0" encoding=" windows-1251"?>
<NEW_ORDER>
  <SHOP_ID>123456789</SHOP_ID>
  <SHOP_PASSWD>paSswoRd</SHOP_PASSWD>
  <AMOUNT>510000</AMOUNT>
  <ORDER_NUMBER>987654321</ORDER_NUMBER>
  <ORDER_DESCRIPTION>Тестовый заказ</ORDER_DESCRIPTION>
  <LANGUAGE>RU</LANGUAGE>
  <BACK_URL>https://pay.avangard.ru</BACK_URL>
  <BACK_URL_OK>https://pay.avangard.ru/thank_you</BACK_URL_OK>
  <BACK_URL_FAIL>https://pay.avangard.ru/order</BACK_URL_FAIL>
  <CLIENT_NAME>Иванов Иван Иванович</CLIENT_NAME>
  <CLIENT_ADDRESS>г. Москва, ул. Садовническая 12</CLIENT_ADDRESS>
  <CLIENT_EMAIL>test@avangard.ru</CLIENT_EMAIL>
  <CLIENT_PHONE>+74951234567</CLIENT_PHONE>
  <CLIENT_IP>127.0.0.1</CLIENT_IP>
  <IS_QR>0</IS_QR>
  <return_qr_image>0</return_qr_image>
  <CARD_NUM>2202654654654646</ CARD_NUM >
  <EXP_YEAR>25</EXP_YEAR>
  <EXP_MONTH>10</EXP_MONTH>
  <CVV>555</CVV>
</NEW_ORDER>
```

Таблица 2.2.2. Выходные параметры для сервиса регистрации заказа.

Имя параметра	Тип	Обязательность для заполнения	Описание параметра
id	Number(10)	+	Идентификатор запроса в АБС банка Авангард
ticket	Varchar(40)	+	Тикет
ok_code	Varchar(10)	+	Код, возвращаемый в случае успешной оплаты
failure_code	Varchar(10)	+	Код, возвращаемый в случае каких-либо ошибок в процессе оплаты
response_code	Number(10)	+	Код ответа для операции регистрации заказа

response_message	Varchar(500)	+	Детальное сообщение ответа
qr_png_base64	Varchar	-	Изображение Qr-кода в Base64 png, будет возвращено, если return_qr_image=1 .

Пример выходного сообщения:

```
<?xml version="1.0" encoding="windows-1251"?>
<order_response>
  <id>123456</id>
  <ticket>1234567890ABCDEABCDE12345678901234567890</ticket>
  <ok_code>TeSt131</ok_code>
  <failure_code>FaiL1213Re</failure_code>
  <response_code>0</response_code>
  <response_message>Успешное выполнение запроса</response_message>
</order_response>
```

Описание кодов ответов находится в приложении 2.

2.3. Получение информации о заказе в системе Интернет-эквайринга

URL: https://pay.avangard.ru/iacq/h2h/get_order_info

Сервис предназначен для получения информации о статусе оплаты заказа в системе Интернет-эквайринга.

Таблица 2.3.1. Входные параметры для сервиса получения информации о заказе.

Имя параметра	Тип	Обязательность для заполнения	Описание параметра
ticket	Varchar(40)	+	Тикет
shop_id	Number(10)	+	Уникальный идентификационный номер торговой точки
shop_passwd	Varchar(32)	+	Пароль для данной торговой точки
version	Number(1)	-	Версия формата ответа, поддерживаются значения от 1 до 4

Входные параметры передаются в формате XML, передача возможна как методом GET, так и POST (предпочтительно), наименование параметра с входящим XML-сообщением – “xml” (без кавычек). Регистр элементов во входящем сообщении и их порядок следования не важен.

Пример входного сообщения:

```
<?xml version="1.0" encoding=" UTF-8"?>
<get_order_info>
  <ticket>1234567890ABCDEABCDE12345678901234567890</ticket>
  <shop_id>123456789</shop_id>
  <shop_passwd>qwertyAAAbb</shop_passwd>
</get_order_info>
```

Таблица 2.3.2. Выходные параметры для сервиса получения информации о заказе.

Имя параметра	Тип	Обязательность для заполнения	Версии	Описание параметра
id	Number(10)	+	1/2/3	Идентификатор запроса в АБС банка Авангард
method_name	Varchar(3)	+	1/2/3	Метод подтверждения платежа. Возможные способы – CVV, D3S, SCR, SMS.*
auth_code	Varchar(10)	+	1/2/3	Код авторизации

status_code	Number(10)	+	1/2/3	Статус заказа
status_desc	Varchar(500)	+	1/2/3	Описание статуса заказа
status_date	Timestamp	+	1/2/3	Дата последнего изменения статуса заказа
response_code	Number(10)	+	1/2/3	Код ответа
response_message	Varchar(500)	+	1/2/3	Детальное сообщение ответа
amount	Number(24)	-	2/3	Сумма заказа
refund_amount	Number(24)	-	2/3	Сумма, возвращенная клиенту
card_num	Varchar(16)	-	2/3	Номер карты, с которой осуществлялась оплата. Отображаются первые 6 и последние 4 цифры.
exp_mm	Number(2)	-	2/3	Месяц окончания срока действия карты
exp_yy	Number(2)	-	2/3	Год окончания срока действия карты
rrn	Varchar(12)	-	3	Ссылка rrn
txn	Varchar(36)	-	4	Номер транзакции (раздел 1.1.3)

** - CVV – подтверждение операции с помощью ввода кода CVV2/CVC2, D3S – подтверждение с помощью механизма 3D Secure (Verified by Visa/MasterCard Secure Code), SCR/SMS – подтверждение с помощью ввода кода со скретч-карты/из SMS-сообщения, данный способ доступен только для карт эмитированных банком Авангард.*

Примеры выходного сообщения:

```
<?xml version="1.0" encoding="UTF-8"?>
<order_info>
  <id>1234567890</id>
  <method_name>SCR</method_name>
  <auth_code>ABC123</auth_code>
  <status_code>5</status_code>
  <status_desc>Авторизация успешно завершена</status_desc>
  <status_date>2012-04-23T12:47:00+04:00</status_date>
  <response_code>0</response_code>
  <response_message>Успешное выполнение запроса</response_message>
</order_info>
```

Описание кодов ответов находится в приложении 2.

2.4. Отмена заказа в системе Интернет-эквайринга

URL: https://pay.avangard.ru/iacq/h2h/reverse_order

Сервис предназначен для отмены заказа в системе Интернет-эквайринга.

Таблица 2.4.1. Входные параметры для сервиса отмены заказа.

Имя параметра	Тип	Обязательность для заполнения	Описание параметра
ticket	Varchar(40)	+	Тикет
shop_id	Number(10)	+	Уникальный идентификационный номер торговой точки
amount	Number(24)	-	Сумма, на которую осуществляется возврат. В случае если параметр не задан, возврату подлежит полная сумма заказа
shop_passwd	Varchar(32)	+	Пароль для данной торговой точки

Входные параметры передаются в формате XML, передача возможна как методом GET, так и POST (предпочтительно), наименование параметра с входящим XML-сообщением – “xml” (без кавычек). Регистр элементов во входящем сообщении и их порядок следования не важен.

Пример входного сообщения:

```
<?xml version="1.0" encoding="windows-1251"?>
<reverse_order>
   <ticket>1234567890ABCDEABCDE12345678901234567890</ticket>
   <shop_id>123456789</shop_id>
   <shop_passwd>qwertyAAAbb</shop_passwd>
</reverse_order>
```

Таблица 2.4.2. Выходные параметры для сервиса отмены заказа.

Имя параметра	Тип	Обязательность для заполнения	Описание параметра
id	Number(10)	+	Идентификатор запроса в АБС банка Авангард
ticket	Varchar(40)	+	Тикет
response_code	Number(10)	+	Код ответа
response_message	Varchar(500)	+	Детальное сообщение ответа

Пример выходного сообщения:

```
<?xml version="1.0" encoding=" windows-1251"?>
<reverse_order_response>
   <id>3535000</id>
   <ticket>1234567890ABCDEABCDE12345678901234567890</ticket>
   <response_code>0</response_code>
   <response_message>Успешное выполнение запроса </response_message>
</reverse_order_response>
```

Описание кодов ответов находится в приложении 2.

2.5. Получение списка операций по номеру заказа

URL: https://pay.avangard.ru/iacq/h2h/get_opers_list

Сервис предназначен для получения полного списка операций по номеру заказа, передаваемого из системы Интернет-магазина. Данный сервис удобно использовать в связке с формированием заказа посредством POST-запроса (см. **1.1.2**), т.к. в данном случае конкретный идентификатор тикета неизвестен, но есть номера заказа в системе Интернет-магазина. Важным нюансом использования является то обстоятельство, что помимо конечного статуса тикета выводятся все оплаты и отмены по нему. Например, если тикет был оплачен, а затем отменен, то в истории операций будет две записи – одна по оплате, а другая по отмене.

Таблица 2.5.1. Входные параметры для сервиса получения списка операций.

Имя параметра	Тип	Обязательность для заполнения	Описание параметра
order_number	Varchar(100)	+	Номер заказа в системе Интернет-магазина
shop_id	Number(10)	+	Уникальный идентификационный номер торговой точки
shop_passwd	Varchar(32)	+	Пароль для данной торговой точки
version	Number(1)	-	Версия формата ответа, поддерживаются значения 1, 2 и 3

Входные параметры передаются в формате XML, передача возможна как методом GET, так и POST (предпочтительно), наименование параметра с входящим XML-сообщением – “xml” (без кавычек). Регистр элементов во входящем сообщении и их порядок следования не важен.

Пример входного сообщения:

```
<?xml version="1.0" encoding="windows-1251"?>
<get_opers_list>
   <order_number>144</order_number >
   <shop_id>123456789</shop_id>
   <shop_passwd>qwertyAAAbb</shop_passwd>
</get_opers_list>
```

Таблица 2.5.2. Выходные параметры для сервиса получения списка операций.

Имя параметра	Тип	Обязательность для заполнения	Описание параметра
oper_info	См. таблицу 2.5.3	-	Вложенная структура с описанием операции. Количество элементов соответствует числу операций.
response_code	Number(10)	+	Код ответа

response_message	Varchar(500)	+	Детальное сообщение ответа

Таблица 2.5.3. Структура данных информации об операции по заказу.

Имя параметра	Тип	Обязательность для заполнения	Версии	Описание параметра
id	Number(10)	+	1/2/3	Идентификатор запроса в АБС банка Авангард
ticket	Varchar(40)	+	1/2/3	Тикет
order_number	Varchar(100)	+	1/2/3	Номер заказа в системе Интернет-магазина
amount	Number(24)	+	1/2/3	Сумма заказа
refund_amount	Number(24)	-	1/2/3	Сумма, возвращенная клиенту
method_name	Varchar(3)	+	1/2/3	Метод подтверждения платежа. Возможные способы – CVV, D3S, SCR, SMS.*
auth_code	Varchar(10)	+	1/2/3	Код авторизации
status_code	Number(10)	+	1/2/3	Статус заказа
status_desc	Varchar(500)	+	1/2/3	Описание статуса заказа
status_date	Timestamp	+	1/2/3	Дата последнего изменения статуса заказа
card_num	Varchar(16)	-	1/2/3	Номер карты, с которой осуществлялась оплата. Отображаются первые 6 и последние 4 цифры.
exp_mm	Number(2)	-	1/2/3	Месяц окончания срока действия карты
exp_yy	Number(2)	-	1/2/3	Год окончания срока действия карты
refund_amount	Number(24)	-	2/3	Итоговая сумма, возвращенная клиенту

refund_amount_part	Number(24)	-	3	Сумма возврата по операции
fee_amount	Number(24)	-	2/3	Сумма комиссии Банка

** - CVV – подтверждение операции с помощью ввода кода CVV2/CVC2, D3S – подтверждение с помощью механизма 3D Secure (Verified by Visa/MasterCard Secure Code), SCR/SMS – подтверждение с помощью ввода кода со скретч-карты/из SMS-сообщения, данный способ доступен только для карт эмитированных банком Авангард.*

Пример выходного сообщения:

```
<?xml version="1.0" encoding="windows-1251" standalone="yes"?>
<opers_list>
  <oper_info>
    <id>1054751</id>
    <ticket>1234567890ABCDEABCDE12345678901234567890</ticket>
    <order_number>1</order_number>
    <status_code>1</status_code>
    <status_desc>Обрабатывается</status_desc>
    <status_date>2013-08-14T10:23:49+04:00</status_date>
    <amount>10000.0</amount>
  </oper_info>
  <oper_info>
    <id>1054752</id>
    <ticket>1234567890ABCDEABCDE12345678901234567811</ticket>
    <order_number>1</order_number>
    <status_code>1</status_code>
    <status_desc>Обрабатывается</status_desc>
    <status_date>2013-08-14T10:24:00+04:00</status_date>
    <amount>10000.0</amount>
  </oper_info>
  <oper_info>
    <id>1054753</id>
    <ticket>1234567890ABCDEABCDE12345678901234567822</ticket>
    <order_number>1</order_number>
    <method_name>CVV</method_name>
    <status_code>2</status_code>
    <status_desc>Отбракован</status_desc>
    <status_date>2013-08-14T10:27:17+04:00</status_date>
```

```
    <amount>10000.0</amount>
    <refund_amount>10000.0</refund_amount>
    <card_num>411111******1111</card_num>
    <exp_mm>12</exp_mm>
    <exp_yy>15</exp_yy>
  </oper_info>
  <response_code>0</response_code>
  <response_message>Успешное выполнение запроса</response_message>
</opers_list>
```

Описание кодов ответов находится в приложении 2.

2.6. Получение списка операций за определенную дату

URL: https://pay.avangard.ru/iacq/h2h/get_opers_by_date

Сервис предназначен для получения полного списка операций за конкретную дату. Ответ сервиса полностью идентичен таковому в сервисе получения списка операций по номеру заказа (**2.5**). Важным нюансом использования является то обстоятельство, что помимо конечного статуса тикета выводятся все оплаты и отмены по нему. Например, если тикет был оплачен, а затем отменен, то в истории операций будет две записи – одна по оплате, а другая по отмене.

Таблица 2.6.1. Входные параметры для сервиса получения списка операций.

Имя параметра	Тип	Обязательность для заполнения	Описание параметра
date	Char(10)	+	Дата в формате dd.mm.yyyy
shop_id	Number(10)	+	Уникальный идентификационный номер торговой точки
shop_passwd	Varchar(32)	+	Пароль для данной торговой точки
version	Number(1)	-	Версия формата ответа, поддерживаются значения 1 и 2

Входные параметры передаются в формате XML, передача возможна как методом GET, так и POST (предпочтительно), наименование параметра с входящим XML-сообщением – “xml” (без кавычек). Регистр элементов во входящем сообщении и их порядок следования не важен.

Пример входного сообщения:

```
<?xml version="1.0" encoding="windows-1251"?>
<get_opers_by_date>
  <date>15.10.2013</date>
  <shop_id>123456789</shop_id>
  <shop_passwd>qwertyAAAbb</shop_passwd>
</ get_opers_by_date >
```

Ответ сервиса, как уже ранее упоминалось, полностью идентичен таковому в сервисе получения списка операций по номеру заказа (см. таблицы **2.5.2** и **2.5.3**).

2.7. Запрет оплаты

URL: https://pay.avangard.ru/iacq/h2h/cancel_order

Сервис предназначен для отмены попытки оплаты. Если по каким-либо причинам клиент не хочет, что бы пользователь смог провести оплату, то он должен вызвать этот сервис.

Таблица 2.7.1. Входные параметры для сервиса получения списка операций.

Имя параметра	Тип	Обязательность для заполнения	Описание параметра
shop_id	Number(10)	+	Уникальный идентификационный номер торговой точки
shop_passwd	Varchar(32)	+	Пароль для данной торговой точки
ticket	Varchar(40)	+	Тикет

Пример входного сообщения:

```
<?xml version="1.0" encoding="windows-1251"?>
<cancel_order>
  <shop_id>123456789</shop_id>
  <shop_passwd>qwertyAAAbb</shop_passwd>
  <ticket>1234567890ABCDEABCDE12345678901234567822</ticket>
</cancel_order >
```

Таблица 2.7.2. Выходные параметры для сервиса получения списка операций.

Имя параметра	Тип	Обязательность для заполнения	Описание параметра
response_code	Number(10)	+	Код ответа
response_message	Varchar(500)	+	Детальное сообщение ответа

Описание кодов ответов находится в приложении 2.

3. Сервис уведомления об успешной оплате/авторизации

3.1. Общая информация о сервисе

Для получения статуса заказа Интернет-магазин может самостоятельно инициировать запрос к эквайрингу (см. разделы **2.3** и **2.5**). Однако в некоторых случаях более удобным вариантом является получение информации в рамках исходящего запроса от самого сервиса Интернет-эквайринга. После ввода всех реквизитов карты и последующего **успешного** завершения оплаты клиентом формируется запрос на уведомление Интернет-магазина. При этом параметры уведомления настраиваются либо при подключении к системе, либо посредством обращения к технической поддержке. В настоящий момент система поддерживает следующие способы уведомления – POST, XML. Адрес, куда эквайринг будет отправлять запрос, настраивается для Интернет-магазина в индивидуальном порядке. Система делает три попытки уведомить магазин о факте успешной оплаты с интервалом в 1 минуту. В случае успешного получения и обработки уведомления Интернет-магазин должен вернуть http-статус ответа сервера **202** (Accepted). Если код ответа сервера не соответствует указанному, либо ответ в принципе не получен, то система делает повторную попытку, но не более трех раз.

Список параметров, которые отправляются Интернет-магазину во время уведомления, приведены в таблице **3.1.1**.

Таблица 3.1.1. Параметры при уведомлении об успешной оплате.

Имя параметра	Тип	Обязательность для заполнения	Описание параметра
id	Number(10)	+	Идентификатор запроса в АБС банка Авангард
ticket	Varchar(40)	+	Тикет
method_name	Varchar(3)	+	Метод подтверждения платежа. Возможные способы – CVV, D3S, SCR, SMS.*
auth_code	Varchar(10)	+	Код авторизации
status_code	Number(10)	+	Статус заказа
status_desc	Varchar(500)	+	Описание статуса заказа
status_date	Timestamp	+	Дата последнего изменения статуса заказа
shop_id	Number(10)	+	Уникальный идентификационный номер торговой точки

order_number	Varchar(100)	+	Уникальный идентификационный номер заказа в системе Вашего Интернет-магазина.
amount	Number(24)	+	Сумма заказа
refund_amount	Number(24)	-	Сумма, возвращенная клиенту
card_num	Varchar(16)	+	Номер карты, с которой осуществлялась оплата. Отображаются первые 6 и последние 4 цифры.
exp_mm	Number(2)	+	Месяц окончания срока действия карты
exp_yy	Number(2)	+	Год окончания срока действия карты
signature	Char(32)	+	Подпись заказа

** - CVV – подтверждение операции с помощью ввода кода CVV2/CVC2, D3S – подтверждение с помощью механизма 3D Secure (Verified by Visa/MasterCard Secure Code), SCR/SMS – подтверждение с помощью ввода кода со скретч-карты/из SMS-сообщения, данный способ доступен только для карт эмитированных банком Авангард.*

Алгоритм формирования контрольной суммы сообщения (signature) аналогичен таковому при формировании заказа POST-запросом с той лишь разницей, что используется не подпись Интернет-магазина, а подпись системы эквайринга. Выглядит это следующим образом:

signature = UPPER(MD5(UPPER(MD5(***av_sign***) + MD5(***shop_id*** + ***order_number*** + ***amount***))))

, где

signature –контрольная сумма, которая передается в качестве одного из параметров;

av_sign – "подпись" системы эквайринга (выдается техподдержкой);

shop_id – идентификатор магазина;

order_number – номер заказа;

amount – сумма заказа (в копейках);

MD5 – шестнадцатеричное представление хеша, сгенерированного по алгоритму MD5;

UPPER – перевод символов в верхний регистр;

Символ «+» (плюс) – конкатенация строк;

Подпись эквайринга (acq_sign) как и в случае подписи магазина не должна предаваться огласке третьим лицам. Она является уникальной для каждого Интернет-магазина. Сервер торговой точки при получении сообщения от системы эквайринга

обязательно должен проверить контрольную сумму для фильтрации возможных запросов злоумышленников.

Как уже говорилось ранее, в качестве способов уведомления поддерживаются POST и XML. В случае метода POST случаях все поля из таблицы **3.1.1** передаются как отдельные параметры. Если же выбран метод уведомления XML, то на сервер Интернет-магазина будет отправлен POST-запрос в кодировке **UTF-8**, где в параметре **xml** будет находиться XML-документ, включающий в себя все необходимые поля заказа (таблица **3.1.1**).

Пример сообщения от сервиса уведомления в формате XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<order_info>
  <id>3535350006</id>
  <ticket>12341411AAA11313131XXX</ticket>
  <shop_id>1234</shop_id>
  <order_number>113-AA</order_number>
  <amount>61500</amount>
  <method_name>CVV</method_name>
  <auth_code>ABC123456</auth_code>
  <status_code>5</status_code>
  <status_desc>Авторизация успешно завершена</status_desc>
  <status_date>2012-04-23T12:47:00+04:00</status_date>
  <signature>5C426D69AA7AC904A9B7C94B42AEB311</signature>
  <card_num>411111******1111</card_num>
  <exp_mm>12</exp_mm>
  <exp_yy>15</exp_yy>
</order_info>
```

3.2. Рекомендации по безопасному использованию сервиса

Немаловажным аспектом настройки взаимодействия с сервисом уведомлений является обеспечение должного уровня безопасности со стороны Интернет-магазина. В силу того, что сервер Интернет-эквайринга сам инициирует запрос к серверу Интернет-магазина, возникает потенциальная возможность компрометации запроса и/или отправка ложного уведомления об оплате. Настоятельно рекомендуется придерживаться следующих правил при реализации сервиса получения уведомлений:

1) **Всегда** проверять подпись заказа (поле ***signature***), полученную от сервера Интернет-эквайринга. Механизм формирования подписи на основе ***av_sign*** подробно описан в разделе **3.1**.
2) По возможности использовать защищенную https часть своего магазина/сайта для приема уведомлений от сервера Интернет-эквайринга во избежание атаки man-in-the-middle.
3) Сохранять максимум информации о факте уведомления об успешной оплате (в частности, IP-адрес с которого осуществлялся запрос).
4) Ограничить доступ к скрипту (странице) приема уведомлений только для IP-адресов серверов Интернет-эквайринга Банка Авангард (актуальный адрес подсети уточняется у техподдержки). Сделать это можно следующим образом:

   При использовании веб-сервера Apache версии 2.4 директория, содержащая скрипт, обрабатывающий запрос от эквайринга, должна быть описана в конфигурационном файле виртуального хоста следующим образом:

   ```
   <Directory "/var/www/path/to/script">
           Require ip АДРЕС_ПОДСЕТИ
   </Directory>
   ```

   , где /var/www/path/to/script – абсолютный путь до директории со скриптом,

   АДРЕС_ПОДСЕТИ – адрес подсети серверов Интернет-эквайринга (уточняется у техподержки).

   Обязательно следует учесть, что доступ ко всей директории будет доступен только для запросов из указанной подсети, поэтому данный скрипт удобнее всего располагать в отдельной директории.

   В качестве альтернативы в директории со скриптом можно создать файл с именем .htaccess со следующим содержимым:

   ```
   Require ip АДРЕС_ПОДСЕТИ
   ```

   В случае использования сервера Apache версии 2.2 конфигурация директории должна быть такой:

   ```
   <Directory "/var/www/path/to/script">
           Order deny,allow
           Deny from all
           Allow from АДРЕС_ПОДСЕТИ
   ```

</Directory>

Или создание файла .htaccess в директории со скриптом:

```
Order deny,allow
Deny from all
Allow from АДРЕС_ПОДСЕТИ
```

При использовании сервера Nginx конфигурация следующая:

```
deny all;
allow АДРЕС_ПОДСЕТИ;
```

Она вносится в соответствующий server или location.

Приложение 1. Статусы заказов, типы данных в системе

Таблица статусов заказов.

Код статуса	Описание статуса
0	Заказ не найден
1	Обрабатывается
2	Отбракован
3	Исполнен
5	Частичный возврат
6	Возврат

Таблица условных обозначений типов данных.

Обозначение	Описание
Number(X)	Целое число, до X знаков
Varchar(X)	Строка, до X знаков
Char(X)	Строка длиной строго X знаков
Timestamp	Дата/время в формате yyyy-MM-dd'T'HH:mm:ssz

Приложение 2. Коды ответов сервисов

Таблица общих кодов ответа для всех сервисов.

Код ответа	Описание ответа
0	Успешное выполнение запроса
1	Поле shop_id пусто
2	Поле shop_passwd пусто
3	Неверное значение в поле shop_id и/или shop_passwd
4	Внутренняя ошибка системы
5	Поле ticket пусто
6	Недопустимый IP адрес. Отправка host2host запросов с данного адреса невозможна
7	Некорректный XML-запрос. Необходимо провалидировать отправляемые данные в соответствии со схемой (см. приложение 2)
8	Пустой xml-запрос. Необходимо убедиться, что сообщение передается внутри параметра с именем xml
9	Неподдерживаемая кодировка запроса. Для XML поддерживаются windows-1251 и UTF-8, для SOAP только UTF-8
10	Некорректный формат суммы

Таблица кодов ошибок специфичных только для сервиса регистрации заказа (2.2)

Код ответа	Описание ответа
101	Поле order_number пусто
104	Поле order_description пусто

105	Поле back_url пусто
106	Поле amount пусто
107	Поле lang пусто
108	Поле shop_passwd пусто

Таблица кодов ошибок, возвращаемых сервисом получения информации о заказе (2.3)

Код ответа	Описание ответа
201	Неверное значение ticket

Таблица кодов ошибок, возвращаемых сервисом отмены заказа (2.4)

Код ответа	Описание ответа
301	Неверное значение ticket
302	Некорректное состояние заказа
303	Отмена заказа невозможна
304	Недопустимая сумма возврата

Таблица кодов ошибок, возвращаемых сервисом получения списка операций по номеру заказа (2.5)

Код ответа	Описание ответа
501	Заказ с указанным номером не найден

Таблица кодов ошибок, возвращаемых сервисом получения списка операций за определенную дату (2.6)

Код ответа	Описание ответа
601	Дата не указана либо формат не соответствует допустимому (dd.mm.yyyy)

Таблица кодов ошибок, возвращаемых сервисом запреты оплаты (2.7)

Код ответа	Описание ответа
701	Невозможно запретить оплату заказа

Приложение 3. XML схема для host2host-запросов

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="get_opers_by_date" type="getOpersByDateRequest"/>

  <xs:element name="get_opers_list" type="getOpersListRequest"/>

  <xs:element name="get_order_info" type="getOrderInfoRequest"/>

  <xs:element name="new_order" type="registrationOrderRequest"/>

  <xs:element name="oper_info" type="operInfo"/>

  <xs:element name="opers_list" type="getOpersListResponse"/>

  <xs:element name="order_info" type="getOrderInfoResponse"/>

  <xs:element name="order_response" type="registrationOrderResponse"/>

  <xs:element name="reverse_order" type="reverseOrderRequest"/>

  <xs:element name="reverse_order_response" type="reverseOrderResponse"/>

  <xs:element name="cancel_order" type="cancelOrderRequest"/>

  <xs:element name="cancel_order_response" type="cancelOrderResponse"/>

  <xs:complexType name="abstractRsObject" abstract="true">
    <xs:sequence/>
  </xs:complexType>

  <xs:complexType name="abstractShopRequest" abstract="true">
    <xs:sequence>
      <xs:element name="shop_id" type="xs:long"/>
      <xs:element name="shop_passwd" type="xs:string"/>
```

```
      <xs:element name="version" type="xs:float" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="registrationOrderResponse">
    <xs:complexContent>
      <xs:extension base="abstractRsObject">
        <xs:sequence>
          <xs:element name="id" type="xs:long"/>
          <xs:element name="ticket" type="xs:string"/>
          <xs:element name="ok_code" type="xs:string"/>
          <xs:element name="failure_code" type="xs:string"/>
          <xs:element name="response_code" type="xs:long"/>
          <xs:element name="response_message" type="xs:string"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="reverseOrderResponse">
    <xs:complexContent>
      <xs:extension base="abstractRsObject">
        <xs:sequence>
          <xs:element name="id" type="xs:long" minOccurs="0"/>
          <xs:element name="ticket" type="xs:string" minOccurs="0"/>
          <xs:element name="response_code" type="xs:long" minOccurs="0"/>
          <xs:element name="response_message" type="xs:string" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="getOpersListRequest">
    <xs:complexContent>
      <xs:extension base="abstractShopRequest">
        <xs:sequence>
```

```
            <xs:element name="order_number" type="xs:string"/>
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="operInfo">
      <xs:complexContent>
        <xs:extension base="abstractRsObject">
          <xs:sequence>
            <xs:element name="id" type="xs:long"/>
            <xs:element name="ticket" type="xs:string"/>
            <xs:element name="order_number" type="xs:string"/>
            <xs:element name="method_name" type="xs:string"/>
            <xs:element name="auth_code" type="xs:string"/>
            <xs:element name="status_code" type="xs:long"/>
            <xs:element name="status_desc" type="xs:string"/>
            <xs:element name="status_date" type="xs:dateTime"/>
            <xs:element name="amount" type="xs:double"/>
            <xs:element name="card_num" type="xs:string" minOccurs="0"/>
            <xs:element name="exp_mm" type="xs:string" minOccurs="0"/>
            <xs:element name="exp_yy" type="xs:string" minOccurs="0"/>
            <xs:element name="fee_amount" type="xs:double" minOccurs="0"/>
            <xs:element name="refund_amount" type="xs:double" minOccurs="0"/>
            <xs:element name="refund_amount_part" type="xs:double" minOccurs="0"/>
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="reverseOrderRequest">
      <xs:complexContent>
        <xs:extension base="abstractTicketRequest">
          <xs:sequence>
            <xs:element name="amount" type="xs:double" minOccurs="0"/>
          </xs:sequence>
```

```
      </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="abstractTicketRequest" abstract="true">
    <xs:complexContent>
      <xs:extension base="abstractShopRequest">
        <xs:sequence>
          <xs:element name="ticket" type="xs:string"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="getOpersListResponse">
    <xs:complexContent>
      <xs:extension base="abstractRsObject">
        <xs:sequence>
          <xs:element ref="oper_info" maxOccurs="unbounded"/>
          <xs:element name="response_code" type="xs:long"/>
          <xs:element name="response_message" type="xs:string"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="getOpersByDateRequest">
    <xs:complexContent>
    <xs:extension base="abstractShopRequest">
        <xs:sequence>
          <xs:element name="date" type="xs:string"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

```
<xs:complexType name="getOrderInfoResponse">
  <xs:complexContent>
    <xs:extension base="abstractRsObject">
      <xs:sequence>
        <xs:element name="id" type="xs:long"/>
        <xs:element name="method_name" type="xs:string"/>
        <xs:element name="auth_code" type="xs:string"/>
        <xs:element name="status_code" type="xs:long"/>
        <xs:element name="status_desc" type="xs:string"/>
        <xs:element name="status_date" type="xs:dateTime"/>
        <xs:element name="response_code" type="xs:long"/>
        <xs:element name="response_message" type="xs:string"/>
        <xs:element name="amount" type="xs:double" minOccurs="0"/>
        <xs:element name="refund_amount" type="xs:double" minOccurs="0"/>
        <xs:element name="card_num" type="xs:string" minOccurs="0"/>
        <xs:element name="exp_mm" type="xs:string" minOccurs="0"/>
        <xs:element name="exp_yy" type="xs:string" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="registrationOrderRequest">
  <xs:complexContent>
    <xs:extension base="abstractShopRequest">
      <xs:sequence>
        <xs:element name="amount" type="xs:decimal"/>
        <xs:element name="is_auth_only" type="xs:int" minOccurs="0"/>
        <xs:element name="back_url" type="xs:string"/>
        <xs:element name="back_url_fail" type="xs:string" minOccurs="0"/>
        <xs:element name="back_url_ok" type="xs:string" minOccurs="0"/>
        <xs:element name="client_address" type="xs:string" minOccurs="0"/>
        <xs:element name="client_email" type="xs:string" minOccurs="0"/>
        <xs:element name="client_ip" type="xs:string" minOccurs="0"/>
        <xs:element name="client_name" type="xs:string" minOccurs="0"/>
        <xs:element name="client_phone" type="xs:string" minOccurs="0"/>
```

```
            <xs:element name="language" type="xs:string"/>
            <xs:element name="order_description" type="xs:string"/>
            <xs:element name="order_number" type="xs:string"/>
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="getOrderInfoRequest">
      <xs:complexContent>
        <xs:extension base="abstractTicketRequest">
          <xs:sequence/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="cancelOrderRequest">
      <xs:complexContent>
        <xs:extension base="abstractTicketRequest">
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="cancelOrderResponse">
      <xs:complexContent>
        <xs:extension base="abstractRsObject">
          <xs:sequence>
            <xs:element name="response_code" type="xs:long"/>
            <xs:element name="response_message" type="xs:string"/>
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:schema>
```