Платежный шлюз eCommerceConnect Gateway.

Интерфейс взаимодействия.

Руководство администратора торговой системы

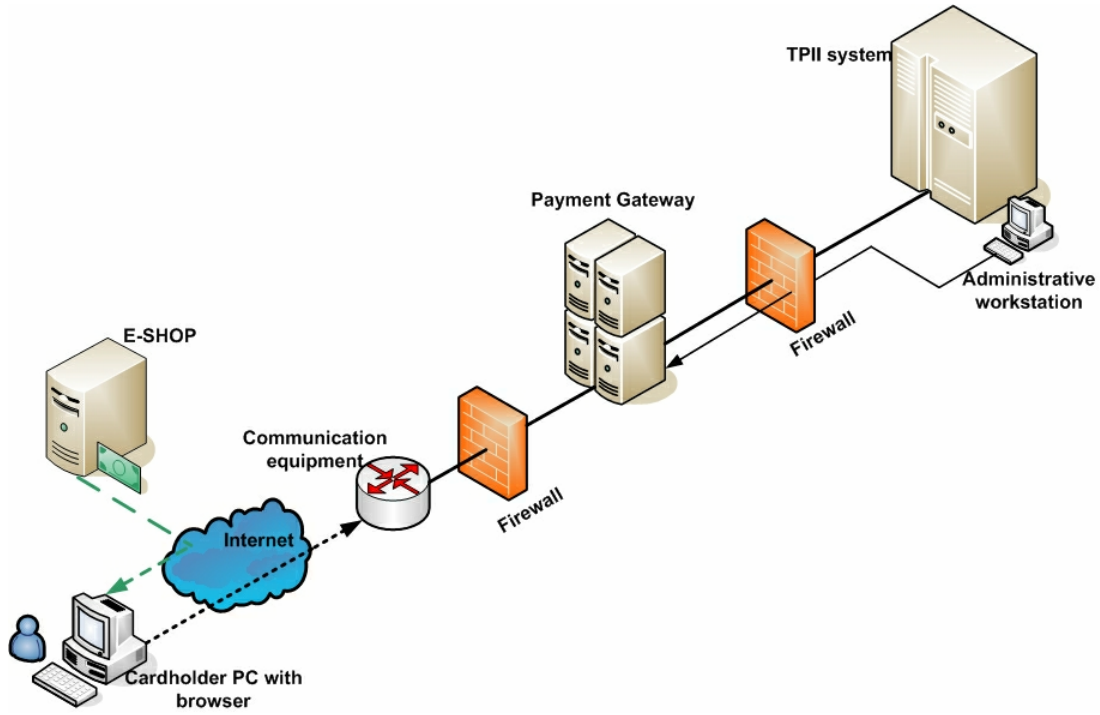
Версия: 3.0

Український процесінговий центр

Содержание

1. Общие положения

Взаимодействие электронного магазина с платежным шлюзом на этапе проверки платежеспособности карты осуществляется на завершающем этапе так называемого ‘checkout’ – процесса. Этот этап, как правило, характерен тем, что покупатель уже определился с набором приобретаемых товаров или услуг, их стоимостью, условиями доставки и т.п. и согласен произвести оплату с использованием платежной карты. Задача торговой системы (электронного магазина) в этот момент - переадресовать покупателя на защищенную страницу платежного сервера, как показано на рисунке, передав при этом необходимые параметры сделки в строке переадресации.



После переадресации на защищенную страницу шлюза взаимодействие с покупателем осуществляется по безопасному протоколу https. Для этого платежный шлюз снабжён SSL-сертификатом, полученным от сертификационного агенства (например, такого как компания “VeriSign”). Однако для аутентификации магазина и защиты данных от модификации в процессе переадресации все критичные данные защищаются с применением MAC (Message Authentication Code).

Программное обеспечение торговой системы (электронного магазина) должно содержать страницы для взаимодействия со шлюзом:

1. Страница с подготовленными значениями для передачи запроса платежному шлюзу

2. Страница (**SUCCESS_URL**)для редиректа браузера пользователя в случае успешного проведения транзакции. В параметрах ответа передаются результаты обработки .

3. Страница (**FAILURE_URL**) для редиректа браузера пользователя в случае неуспешного проведения транзакции. В параметрах ответа передаются результаты обработки .

4. Страница (**NOTIFY_URL**) для передачи результата транзакции от шлюза напрямую электронному магазину (опционально) .

Если данная страница не используется, то все результаты обработки передаются через страницу браузера на адрес магазина (п. 2, 3) . Применение же данной страницы позволяет напрямую передать результаты транзакции от шлюза торговцу. Таким образом повышается уровень безопасности - торговец доверяет соединению со стороны шлюза, так как адрес такого источника фиксирован, в отличие от адреса браузера покупателя. Кроме того, после такого подтверждения, при редиректе покупателя (п. 2, 3) , в параметрах ответа передается только некритичная часть результатов обработки, обеспечивая, таким образом, скрытие наиболее критичных данных от покупателя.

Часть программного обеспечения использует динамические элементы при формировании URL. Зачастую это происходит, если серверное программное обеспечение или браузер не поддерживают или отключают схему поддержки cookies. В таком случае торговец должен предоставить схему формирования URL .

2. Передача параметров авторизационного запроса платежному серверу.

Торговая система (электронный магазин) должна передать ряд параметров при переходе на защищенную страницу шлюза. Эти параметры приведены в таблице 1:

Таблица 1

Параметр	Структура	Формат	Наменование (назначение) параметра	Доп. комментарий
Version	F	N	Значение версии интерфейса SG	Версия протокола взаимодействия. Действующая версия в данный момент '1'. Этот параметр является справочным для обработчика входных данных шлюза.
MerchantID	L	an15	Идентификатор торговца	Назначается обслуживающим банком
TerminalID	F	an8	Идентификатор терминала	-- // --
TotalAmount	F	N1..12	Сумма заказа	В мелких единицах валюты (копейки, центы)
Currency	F	n3	Валюта	Определяется договором с обслуживающим банком
AltTotalAmount (O)	F	N1..12	Сумма заказа (альтернативная валюта)	Необязательный параметр В мелких единицах валюты (копейки, центы)
AltCurrency (O)	F	n3	Альтернативная Валюта	Определяется, если магазин желает отобразить сумму платежа в другой валюте.
PurchaseTime	F	n12..17	Время запроса в формате yyMMddHHmmss или yyMMddHHmmssZ	yy - year MM - month in year dd - day in month HH - hour in day (0-23) mm - minute in hour ss - second in minute Z - time zone (RFC 822) Формат зоны - [+|-] Hours Minutes Например +0300 , -0200 Если в параметре зона не указана, считается та же зона, что и у шлюза
locale	F	a2	Язык интерфейса (en, ru, uk)	Язык интерфейса защищенной страницы шлюза
OrderID	L	Ans…20	Номер заказа длиной до 20 байт	
SD (O)	Var	an...99	Session Data -Данные сессии	Вспомогательный параметр, может быть использован торговой системой для управления пользовательскими сессиями
PurchaseDesc (O)	L	ans...125	Краткое описание покупки, товара	Необязательный параметр, предусмотренный спецификацией 3-D Secure. Может быть отображен на странице платежного сервера. Рекомендуется использовать для удобства покупателя.
Signature	Var	Зависит от схемы	Значение MAC-кода или подписи	Длина параметра зависит от выбранной схемы вычисления

Delay	F	N1	Идентификатор платежа Пре-Авторизация	Для пре-авторизации, значение должно быть равно «1», иначе 0 или пусто

Примечание 1:

A. Structure description

F – full field
L – left justified
R – right justified
S – filled with spaces
Z – filled with zeroes
Var – поле переменной длины

B. Format description

n- numeric decimal digit, value 0..9,
an - alphabetic or numeric character, value 0..9 or A..Z or ..z,
ans - alphabetic, numeric or special character,

Примечание 2:

Параметры AltTotalAmount, AltCurrency определяются, если продавцу нужно показать в магазине сумму платежа в другой валюте, которая отличается от валюты в договоре обслуживающего банка.
При этом на шлюз нужно посылать 4 параметра:

TotalAmount, Currency - сумма и валюта по условиям договора

AltTotalAmount , AltCurrency - сумма и валюта , указанная для оплаты на магазине.

Коды валют:
643 Russian Ruble
840 United States Dollar
978 Euro
980 Ukrainian Hryvnia

Однако транзакция будет производиться по параметрам TotalAmount, Currency . При этом торговец сам отвечает за соответствие сумм между разными валютами (правильный пересчет в соответствии с курсом).

Данные параметры передаются методом HTTPS/POST на страницу шлюза в определенной HTML-форме для дальнейшего ввода реквизитов платежной карты покупателем (держателем платежной карты).

Пример:

```
<FORM  ACTION="https:/secure.upc.ua/ecgtest/enter" METHOD="POST">

<INPUT TYPE=”HIDDEN” NAME=”Version” VALUE="1">
<INPUT TYPE="HIDDEN" NAME="MerchantID" VALUE="6352045">
<INPUT TYPE="HIDDEN" NAME="TerminalID" VALUE="ECI62791">
<INPUT TYPE="HIDDEN" NAME="TotalAmount" VALUE="12550">
<INPUT TYPE="HIDDEN" NAME="Currency" VALUE="980">
<INPUT TYPE="HIDDEN" NAME="locale" VALUE="ru">
<INPUT TYPE="HIDDEN" NAME="SD" VALUE="584sds565hgj76GGjh6756248">
<INPUT TYPE="HIDDEN" NAME="OrderID" VALUE="HV-923452">
<INPUT TYPE="HIDDEN" NAME="PurchaseTime" VALUE="031227105500">
<INPUT TYPE="HIDDEN" NAME="PurchaseDesc" VALUE="Musical Disc ">
<INPUT TYPE="HIDDEN" NAME="Signature" VALUE="45F345FAFDE4455445AC">

</FORM>
```

Далее на странице шлюза принятые данные дополняются значениями CardType, CardNumber, ExpYear, ExpMonth, CVV2 (при необходимости). Предварительно шлюз производит ряд проверок (наличие регистрационных параметров торговца в БД, соответствие типа валюты зарегистрированному значению, величину авторизационного лимита для торговца, а также проверку электронной подписи).

Затем этапе шлюз отображает в браузере покупателя страницу для ввода реквизитов карты. При этом покупатель также может указать тип карты (при условии, что торговцу разрешено принимать тот или иной тип). Покупатель также может ввести код CVV2 своей карты (для карт MAESTRO ввод данного кода не предлагается)

На следующем этапе происходит обработка запроса с использованием схемы 3D-Secure или стандартной схемы (channel encryption e-commerce), в зависимости от параметров, предоставленных обслуживающим банком.

3. Возврат результатов обработки авторизационного запроса торговой системе.

Перечень параметров ответа сайту торговца:

Таблица 2

Параметр	Формат	Наименование (назначение) параметра	Доп. Комментарий
MerchantID	an15	Идентификатор торговца	Совпадает с данными в запросе авторизации
TerminalID	an8	Идентификатор терминала	--- // ---
TotalAmount	n..12	Сумма заказа	--- // ---
AltTotalAmount	n..12		
Currency	n3	Валюта	--- // ---
AltCurrency	n3		
PurchaseTime	N12	Время запроса сделки (YYMMDDhhmmss)	--- // ---
OrderID	ans..20	Order ID	
XID	ans28	Идентификатор транзакции (номер заказа, дополненный до 20 символов)	--- // ---
SD	an... 99	Данные сессии	--- // ---
ApprovalCode	An6	Код авторизации хоста	
Rrn	N12	Retrieval Reference Number	Уникальный номер транзакции в системе авторизации и расчетов обслуживающего банка
ProxyPan	N13…19	4 последние цифры номера карты.	Значение PAN (4 последние цифры) с дополненными нулями впереди для индикации длины PAN.
TranCode	N3	Код завершения транзакции	См. табл. 3
Signature	An…40	Значение MAC-кода для выбранной схемы взаимодействия шлюза-магазин	Длина параметра зависит от выбранной схемы вычисления MAC-кода
Delay	N1	Идентификатор платежа Пре-Авторизация	

Результаты обработки (завершения транзакции) могут быть переданы двумя способами :

- ***Доставка ответа в странице браузера***

В этом случае результаты передаются через страницу браузера , где соответствующая форма передается по адресу сайта торговца на страницу ‘успешно/ не успешно’ . Действие запуска формы производится с помощью Java Script. При невозможности выполнения данного скрипта (если поддержка Java Script в браузере отключена) выводится сообщение о необходимости вручную подтвердить отправку формы.

Адреса этих страниц электронного магазина извлекаются шлюзом из своей БД, т.е. должны быть предоставлены торговцем заранее – на этапе регистрации.

Пример :

```
<FORM    NAME=”back”  ACTION=”http://www.playboy.kiev.ua/shop/success.asp”
METHOD="POST">

<INPUT TYPE=”HIDDEN” NAME=”Version” VALUE="1">
<INPUT TYPE="HIDDEN" NAME="MerchantID" VALUE="6352045">
<INPUT TYPE="HIDDEN" NAME="TerminalID" VALUE="ECI62791">
<INPUT TYPE="HIDDEN" NAME="TotalAmount" VALUE="12550">
<INPUT TYPE="HIDDEN" NAME="Currency" VALUE="980">
<INPUT TYPE="HIDDEN" NAME="SD" VALUE="584sds565hgj76GGjh6756248">
<INPUT TYPE="HIDDEN" NAME="OrderID" VALUE="VHS-23684">
<INPUT TYPE="HIDDEN" NAME=”XID" VALUE="00005021611-000064-1">
<INPUT TYPE="HIDDEN" NAME="ApprovalCode" VALUE="554632">
<INPUT TYPE="HIDDEN" NAME="Rrn" VALUE="775333567770">
<INPUT TYPE="HIDDEN" NAME="ProxyPan" VALUE="0000000000005207">
<INPUT TYPE="HIDDEN" NAME="TranCode" VALUE="000">
<INPUT TYPE="HIDDEN" NAME="Signature" VALUE="45F345Fafde4455445Gvb550">

</form>

<noscript>
<center>
<h1>Return processing results</h1>
<h2>Your browser noes not support JavaScript or disabled</h2>
<h3>Click 'Submit' to continue with  Transaction</h3>
<input type="submit">
</center>
</noscript>

</form>

<script language="javascript">
<!--
document.back.submit();
-->
</script>
```

Для привязки покупателя к соответствующей сессии торговой системы (электронного магазина) и конкретной покупке используется параметр SD (Session Data), передаваемый через браузер покупателя в процессе обратной переадресации.

Пример:

```
<INPUT TYPE="HIDDEN" NAME="SD" VALUE="584sds565hgj76GGjh6756248">
```

Следует отметить, что успешная и гарантированная переадресация браузера с параметрами результата оплаты является необходимым условием для получения магазином факта оплаты. В ряде случаев, однако, это может плохо сработать при разного рода ситуациях, таких как:

1) сбой работы браузера, зависание;

2) неадекватное поведение пользователя в момент передачи ответа;

3) пропадание соединения с интернет-провайдером

4) неправильная работа браузера при установленных пакетах защиты, которые могут вносить изменения в работу браузера.

Может произойти ситуация, когда оплата карточкой произведена, но магазину результат не доставлен. При этом потребуется разбирательство покупателя с магазином по факту проведения блокировки суммы денег с дальнейшей установкой состояния «Оплачено» либо с проведением возврата.

В таких случаях магазину рекомендуется внедрение схемы с доставкой ответа со стороны шлюза.

Данный метод передачи ответа магазину является предпочтительным и настоятельно рекомендуется. Его использование позволяет существенно уменьшить количество некорректно завершенных транзакций (например, из-за сбоев браузера покупателя или его неверных действий). Даже в случае возникновения проблемных ситуаций магазин будет иметь достоверную информацию о результате обработки транзакции в банке.

- ***доставка ответа по адресу NOTIFY_URL***

Для начала работы с данной схемой необходимо в интерфейсе администратора в разделе терминалы прописать адрес кура будет осуществляется отправка данных сообщений.

В этом случае результаты обработки передаются методом HTTP/HTTPS POST со стороны шлюза на страницу магазина.

Notify request message:
PurchaseTime = '090929152500'
ProxyPan = '499999*******0011'
Currency = '980'
ApprovalCode = '111111'
MerchantID = '1752493'
OrderID = '111111111111111111'
Signature = test'
Rrn = '2222222222'
XID = '333333-4444444'
Email = 's.sichnoy@upc.ua'
SD = '24ee6084a5343e3d'
TranCode = '000'
TerminalID = 'E7880293'
TotalAmount = '500'

При этом магазин возвращает ответ в теле выполняемой страницы. Каждый параметр и его значение в виде Параметр=Значение возвращаются на новой строке.
Строки разделяются символом разделителя строк.

В ответе, дополнительно к оригинальным значениям параметров (MerchantID, TerminalID, OrderID, Currency, TotalAmount, XID, PurchaseTime), возвращает 3 новых параметра –

Параметр	Значение	Описание
Response.action	approve | reverse	При значении 'approve' ел. магазин дает одобрение успешности операции. При значении 'reverse' шлюз делает откат успешной транзакции и устанавливает признак завершения 503 – «Транзакция отменена магазином»
Response.reason	An.. 255	Пояснение ответа магазина (опционально), например – причина для значения Response.action
Response.forwardUrl	An.. 255	Значение URL для редиректа браузера пользователя, вместо SUCCESS_URL или NOTIFY_URL (предоставляет возможность использовать динамические ссылки)

Пример :

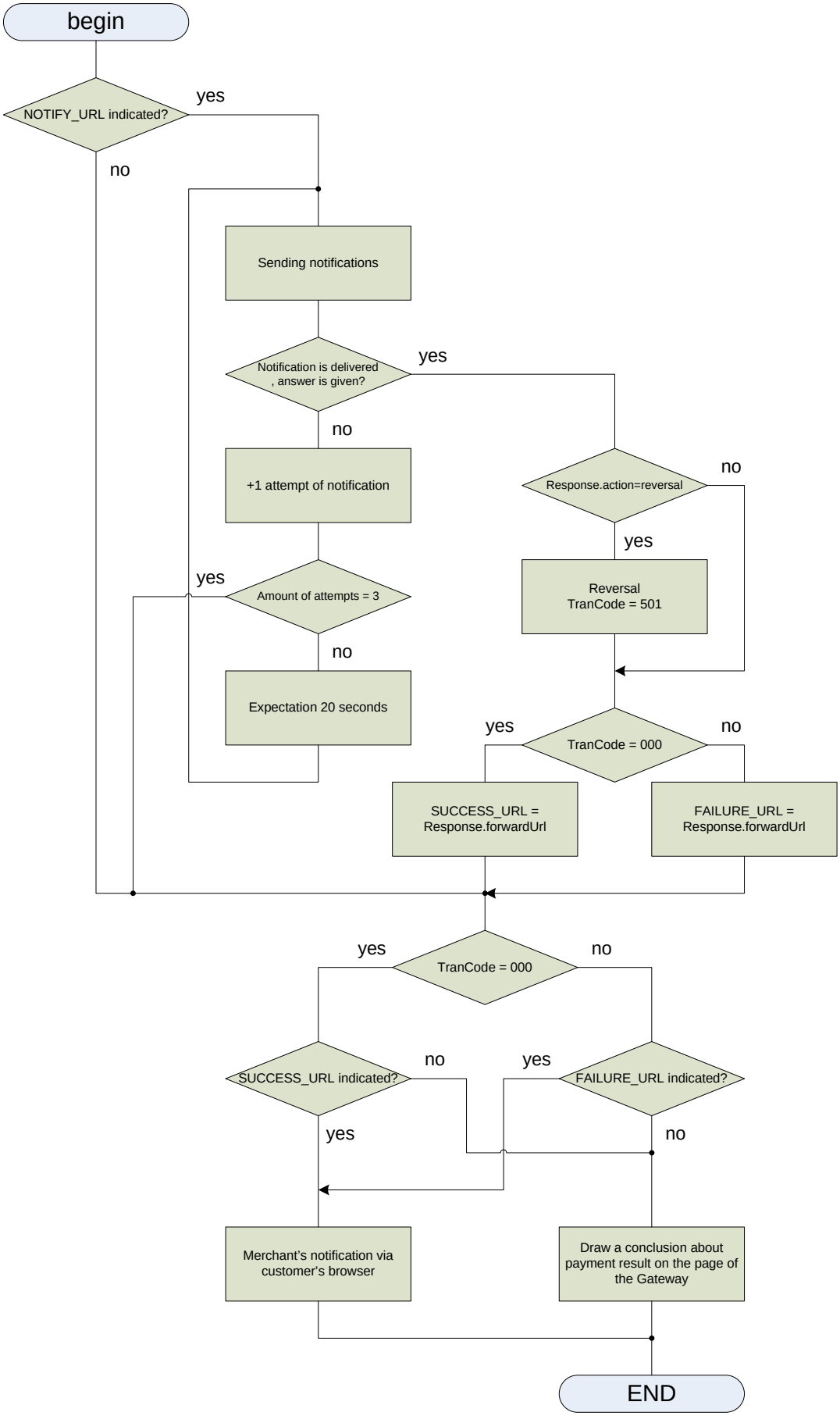
```
echo "MerchantID="1752493"\n";
echo "TerminalID="E7880293"\n";
echo "OrderID="ID0009992"\n";
echo "Currency="980"\n";
echo "TotalAmount="980"\n";
echo "XID="333333-4444444"\n";
echo "PurchaseTime="090929152500"\n";
echo "Response.action=\n";
echo "Response.reason=\n";
echo "Response.forwardUrl=\n;
```

Также у администратора магазина есть выбор разрешать транзакции по которым не проходит доставка сообщения по NOTIFY_URL, либо запретить их. Данная схема позволяет автоматизированным системам учёта предоставленных услуг, которые используют информацию от платежного шлюза, избежать несоответствия данных в случае недоставки сообщения по NOTIFY_URL.

В случае если по каким то причинам схема доставки ответа со стороны шлюза не отработала - транзакция будет автоматически отменена.

В данном случае оригинальная транзакция имеет тип операции покупка и код транзакции 504/Данный запрос на оплату не разрешен платежным шлюзом, откат произведенный в автоматическом режиме платежным сервером имеет код транзакции 000/Транзакция завершена успешно.

Далее представлена логика доставки ответа со стороны шлюза.


begin
NOTIFY_URL indicated?
yes
no
Sending notifications
Notification is delivered , answer is given?
yes
no
+1 attempt of notification
Response.action=reversal
no
yes
Reversal TranCode = 501
Amount of attempts = 3
yes
no
Expectation 20 seconds
TranCode = 000
yes
no
SUCCESS_URL = Response.forwardUrl
FAILURE_URL = Response.forwardUrl
TranCode = 000
yes
no
SUCCESS_URL indicated?
no
yes
FAILURE_URL indicated?
yes
no
Merchant's notification via customer's browser
Draw a conclusion about payment result on the page of the Gateway
END

4. Коды возврата результата транзакции

Коды возврата результата транзакции делятся на несколько классов и подклассов и служат для информирования торговой системы о результате ее проведения.
Для индикации успешного завершения транзакции достаточно одного кода. Большая же часть кодов служит для предоставления торговой системе обобщенной информации о причине неудачного завершения транзакции.

Таблица 3

Коды на основе ответов авторизацинного хоста банка		Комментарий
Обобщенные коды ответа для магазина	Примерная интерпретация ответа на странице возврата электронного магазина	Коды ответов в сообщении 1110
000	Сделка авторизована	00x
105	Транзакция не разрешена банком-эмитентом	100, 103,104,105…107,
116	Недостаточно средств	116
111	Несуществующая карта	111,125,200,202
108	Карта утеряна или украдена	208,209
101	Неверный срок действия карты	101,201
130	Превышен допустимый лимит расходов	121,123
290	Банк-издатель недоступен	905…908,910
291	Техническая или коммуникационная проблема	9xx (кроме указанных выше)
Коды на основе ответов генерируемых платежным сервером без обращения к хосту банка		
Внутренние коды ошибок платежного сервера в соответствии со способом обработки		
401	Ошибки формата	
402	Ошибки в параметрах Acquirer/Merchant	
403	Ошибки при соединении с ресурсом платежной системы (DS)	
404	Ошибка аутентификации покупателя	
405	Ошибка подписи	
406	Превышена квота разрешенных транзакций	
407	Торговец отключен от шлюза	
408	Транзакция не найдена	
409	Несколько транзакций найдено	
410	Заказ уже был успешно оплачен	В случае повтора
411	Некорректное время в запросе	
412	Параметры заказа уже были получены ранее	В случае повтора
420	Превышен лимит дневной лимит транзакций	
421	Превышена максимально разрешенная сумма транзакции	
430	Транзакция запрещена на уровне платежного шлюза	
431	Не разрешена транзакция без полной аутентификации по схеме 3-D Secure	Карта не зарегистрирована для 3-D Secure операций
501	Транзакция отменена пользователем	
502	Сессия браузера устарела	
503	Транзакция отменена магазином	
504	Транзакция отменена шлюзом	
601	Транзакция не завершена	

5. Запрос состояния транзакции со стороны торговца

Для получения статуса оплаты торговой точке рекомендутся использовать схему с применением NOTIFY_URL. В данном случае шлюз попытается доставить результат на магазин напрямую, не надеясь на передачу параметров через браузер пользователя.
Дополнительно, магазин со своей стороны может послать запрос о состоянии транзакции на шлюз.
Это может быть сделано передачей запроса POST на страницу шлюза со следующими параметрами –

MerchantID=
TerminalID=
OrderID=
Currency=
TotalAmount=
PurchaseTime=

Шлюз возвращает текстовую страницу с дополнительными параметрами –

XID=
TranCode=
ApprovalCode=

Транзакция считается успешной, если значение поля TranCode = “000”.

Данный механизм доставки результатов авторизации является опциональным, но может быть использован в случае проблем с доставкой результатов через броузер покупателя.

Пример:

```
<html>
<body>
<form method='POST' action="https://secure.upc.ua/ecgtest/service/01">
<input type='hidden' name='MerchantID' value='6352045'>
<input type='hidden' name='TerminalID' value='ECI62791'>
<input type='hidden' name='OrderID' value='VHS-23684'>
<input type='hidden' name='Currency' value='980'>
<input type='hidden' name='TotalAmount' value='12550'>
<input type='hidden' name='PurchaseTime'value='031227105500'>
<input type='submit' value='go'>
</form>
</body>
<html>
```

6. Подпись данных .

Подпись применяется для защиты данных оплаты при передаче их к шлюзу и от шлюза. Это дает возможность обнаружить изменение в данных, если они были произведены преднамеренно и, таким образом, защититься от вмешательства.

Платежный шлюз, приняв параметры запроса от торговца, проверяет данные на предмет целостности. Для этого он производит операцию проверки подписи (MAC-кода) торговца. Для этого торговец должен сформировать значение этого MAC-кода на своей стороне а также прислать его как значение параметра 'Signature' .

Подпись формируется путем выполнения криптографической операции с хешем данных и ключем (секретным или приватным) .

Шлюз использует несколько способов для наложения/проверки подписей:

- с использованием аппаратного токена (ридера с чип-картой)
- криптография с помощью публичных ключей

- ***Использование аппаратного токена***

В качестве устройства используется токен (PKCS#11) или кард-ридер с апплетом, записанным в чип карты.

Для формирования подписи используется криптографическая функция токена, устанавливаемого торговцем при регистрации совместно с драйвером. Драйвер токена принимает запрос соответствующего формата и формирует результат с кодом завершения и значением подписи, если код завершения успешный.

Драйвер токена принимает запросы по протоколу TCP/IP и формирует результат.

- ✓ подпись

Форматы запроса на подпись и ответа представляют собой следующее :

ЗАПРОС		
Данные	Формат	Описание
'M0'	an2	Команда токена
'|'	an1	Символ разделителя '|'
'MerchantID'	an…15	Идентификатор торговца, присвоенный при регистрации
'|'	an1	Символ разделителя '|'
Параметры запроса	an…99	Параметры запроса, соединенные знаком ';' , в следующей последовательности : • MerchantID • TerminalID • PurchaseTime • OrderID + (, Delay) • Currency + (, AltCurrency) • TotalAmount + (, AltTotalAmount) • SD Если значение параметра пустое, то оно все равно используется Если используются параметры AltCurrency, AltTotalAmount, то они указываются вместе с полями Currency, TotalAmount соответственно, указав через запятую Например 980,840 - коды валюты 1000,200 - суммы
ОТВЕТ		
'M1'	an2	Команда ответа токена
'|'	an1	Символ разделителя '|'
Код завершения	N2	Код завершения : • 00 - успешно, подпись сформирована • 01 - ошибка, запрос неправильно сформирован • 02 - ошибка, на токене нет объекта, соответствующего данному торговцу • 03 - ошибка выполнения криптографической функции
'|'	an1	Символ разделителя '|'
Подпись	An…8	Значение подписи, шестнадцатиричное значение

Пример : Merchant ID = 6352045

- Запрос:

M0|6352045|6352045;ECI62791;031227105500;HV-923452;;980;12550;584sds565hgj76GGjh6756248;

- Ответ:

M1|00|A878B978

- ✓ Проверка подписи

Платежный сервер, завершив обработку транзакции, формирует ответный результат и его контрольный MAC-код.
Целостность данных, полученых от платежного сервера, можно проверить с помощью команды M2 .

ЗАПРОС		
Данные	Формат	Описание
'M2'	an2	Команда токена
'|'	an1	Символ разделителя '|'
'MerchantID'	an…15	Идентификатор торговца, присвоенный при регистрации
'|'	an1	Символ разделителя '|'
Параметры запроса	an…99	Параметры запроса, соединенные знаком ';' , в следующей последовательности : • MerchantID • TerminalID • PurchaseTime • OrderID + (, Delay) • XID • Currency + (, AltCurrency) • TotalAmount + (, AltTotalAmount) • SD • TranCode - код завершения транзакции • ApprovalCode - код авторизации
'|'	an1	Символ разделителя '|'
подпись	An…32	Значение подписи, шестнадцатиричное значение
ОТВЕТ		
'M3'	an2	Команда ответа токена
'|'	an1	Символ разделителя '|'
Код завершения	N2	Код завершения : • 00 - подпись проверена успешно • 01 - ошибка, запрос неправильно сформирован • 02 - ошибка, на токене нет объекта, соответствующего данному торговцу • 03 - ошибка выполнения криптографической функции

Пример : Merchant ID = 6352045

- Запрос

M2|6352045|6352045;ECI62791;031227105500;HV923452;;980;12550;584sds565hgj76GGjh6756248;000;523453;

- Ответ

M3|00

- ***Использование публичных ключей***

Для начала работы необходимо скачать Win32 OpenSSL. При написании данного руководства использовалась версия Win32 OpenSSL v0.9.8e.

После установки необходимо прописать в переменную Path путь к каталогу bin:
правой кнопкой "Мой компьютер", затем "Свойства", "Дополнительно", "Переменные среды"
В системных переменных : переменная PATH, "Изменить"
поставить в конце строки точку с запятой и прописать путь к папке bin: c:\OpenSSL\bin

Генерация ключей

Генерация и обмен ключами осуществляется после подачи заявки на регистрацию и получения атрибутов Интернет магазина, в частности MerchantID.

Перед генерацией ключей необходимо отредактировать config.dat в соответствии с данными подключаемого магазина.

#Страна
CN=UA
#Область
ST=Kievskaya
#Город
L=Kiev
#Название организации
O=UPC
#Название отделения
OU=IT DEP.
#Имя для сертификата (Ваше имя)
CN=SERG
#Email организации
emailAddress=ec@upc.ua

Данные в config.dat не должны строго соответствовать данным в заявке на подключение, они не участвуют в генерации или проверки подписи, и используются только для идентификации файла сертификата.

Команда run.bat:
openssl genrsa -out %1.pem 1024
openssl req -new -key %1.pem -x509 -days 365 -config config.dat -out %1.crt
openssl x509 -in %1.crt -noout -pubkey > %1.pub
с параметром MerchantID (например, run.bat 1770000)
генерирует три файла:
1770000.pem – приватный ключ
1770000.pub – публичный ключ
1770000.crt – сертификат
Файл сертификата (*.crt) необходимо отправить в UPC, для проверки шлюзом аутентичности сообщения от Интернет магазина.

Генерация подписи

Подпись генерируется на основании двух файлов: *.pem и datafile.
Datafile содержит данные (поля), для которых непосредственно формируется подпись.

Важно соблюдать последовательность полей, иначе запрос будет отклонен с ошибкой 405 (Signature is invalid).
Поля записываются в datafile в таком порядке (в таком же порядке они должны быть и при программной реализации):

MerchantId;TerminalId;PurchaseTime;OrderId,Delay;CurrencyId,AltCurrencyId;
Amount,AltAmount;SessionData(SD);

Количество знаков ; должно оставаться постоянным. Если какое то поле отсутствует, то ставится ;;. Например, если отсутствует SessionData(SD), то datafile будет выглядеть следующим образом:

MerchantId;TerminalId;PurchaseTime;OrderId,Delay;CurrencyId,AltCurrencyId;
Amount,AltAmount;;

Если отсутствуют поля Delay или AltCurrency, AltAmount то запятая перед этими полями опускается. Например:

MerchantId;TerminalId;PurchaseTime;OrderId;CurrencyId,AltCurrencyId;Amount,
AltAmount;;

MerchantId;TerminalId;PurchaseTime;OrderId,Delay;CurrencyId;Amount;;

MerchantId;TerminalId;PurchaseTime;OrderId;CurrencyId;Amount;;

Для правильной генерации подписи необходимо что бы datafile не содержал лишних символов (пробелы, символы перевода каретки и возврата в начало строки).
Необходимо проверить отсутствие лишних символов в HEX редакторе или в FARe (F3, F4). Точно так же эти данные должны быть представлены при программной реализации подписи.

Для генерации подписи необходимо запустить create_signature.bat с параметром *.pem. Например create_signature.bat 1770000.pem.
В результате будут обновлены или созданы два файла: signature.bin (подпись) и signature (подпись в кодировке base64). Данные в файле signature отправляются в запросе как подпись. (Важно. В запросе нельзя писать название полей в нижнем регистре, то есть, поле с именем merchantid таковым не является).

Прверка подписи от шлюза

Для проверки подписи данных со шлюза необходимо в файл from_gateway записать поля в следующем порядке:

MerchantId;TerminalId;PurchaseTime;OrderId,Delay;Xid;CurrencyId,AltCurrencyId;
Amount,AltAmount;SessionData;TranCode;ApprovalCode;

Все требования с предыдущего пункта, касательно формирования from_gateway здесь так же обязательны.

Для проверки подписи со шлюза ее необходимо поместить в файл signature.

Очень важно, что бы количество значащих символов в одной строке файла signature не превышало 64 (длина строки).

Для проверки подписи выполнить check_signature.bat

openssl base64 -d -in signature -out signature.bin
openssl dgst -sha1 -verify test-server.pub -signature signature.bin from_gateway

- ***Примеры***

Пример на PHP :

✓ Формироване запроса на шлюза

```
подготовить данные
?php
$OrderID = 12;
$PurchaseTime = date("ymdHis") ;
$data = "1752429;E7880229;$PurchaseTime;$OrderID;980;1200;;" -строка для
подписи
$fp = fopen("1752429.pem", "r");
$priv_key = fread($fp, 8192);
fclose($fp);
$pkeyid = openssl_get_privatekey($priv_key);
openssl_sign( $data , $signature, $pkeyid);
openssl_free_key($pkeyid);
$b64sign = base64_encode($signature) ;Подпись данных в формате base64
?>
<form action="https://secure.upc.ua/ecgtest/enter" method="post" > -URL
тестового сервера
<input name="Version" type="hidden" value="1" />
<input name="MerchantID" type="hidden" value="1234567" />-Тестовый MerchantId
<input name="TerminalID" type="hidden" value="E1234567" />-Тестовый TerminalID
<input name="TotalAmount" type="hidden" value="242" />
<input name="Currency" type="hidden" value="980" />
<input name="locale" type="hidden" value="RU" />
<INPUT TYPE="HIDDEN" NAME="PurchaseTime" VALUE="<?php echo $PurchaseTime ?>">
<input name="OrderID" type="hidden" value="12" />
<input name="Signature" type="hidden" value="<?php echo "$b64sign" ?>"/>
-Подпись которую сервер проверит Вашим публичным ключем
<input type="submit"/>
</form>
<a href="index112.php">SUBMIT</a>
</body>
</html>
```

- ✓ Проверка ответа ключом шлюза

```
<?php

// $data содержит значение полей для проверки подписи

$signature = $HTTP_POST_VARS["Signature"];
$signature = base64_decode($signature) ;

// извлечь сертификат
$fp = fopen("/opt/deploy/certs/server.crt", "r");
$cert = fread($fp, 8192);
fclose($fp);
$pubkeyid = openssl_get_publickey($cert);

// проверка подписи
$ok = openssl_verify($data, $signature, $pubkeyid);
if ($ok == 1) {
   echo "good";
} elseif ($ok == 0) {
   echo "bad";
} else {
   echo "ugly, error checking signature";
}
// free the key from memory
openssl_free_key($pubkeyid);
?>
```

Пример на Perl :

- ✓ Подпись запроса ключем торговца

```
use Crypt::OpenSSL::RSA;
use MIME::Base64 ;

$data = "$MerchantID;$TerminalID;$PurchaseTime;ORDER-2;980;1200;;" ;

$key_string = `cat /opt/deploy/certs/3333333.pem` ;
$rsa_priv = Crypt::OpenSSL::RSA->new_private_key($key_string);
print "private key is:\n", $rsa_priv->get_private_key_string();

$sign = $rsa_priv->sign($data);
$b64sign = encode_base64($sign) ;

print $b64sign ;
```

- ✓ Проверка ответа ключом шлюза

```
use Crypt::OpenSSL::RSA;
use Crypt::OpenSSL::X509;

use MIME::Base64 ;

#  прочитать/сформировать ключ шлюза
my $x509 = Crypt::OpenSSL::X509->new_from_file( '/sites/certs/server.cert' );
$rsa_pub = Crypt::OpenSSL::RSA->new_public_key($x509->pubkey);
print STDERR "pubkey = " . $x509->pubkey() . "\n";

#  проверка подписи
$sign = decode_base64($b64sign);
$result = $rsa_pub->verify($data,$sign);
```

- ✓ Подпись запроса приватным ключем торговца

Файл **datafile** содержит данные для подписи, например

6352045;ECI62791;031227105500;HV-923452;;980;12550;;

Команда

```
cat datafile | openssl dgst -sha1 -sign 1751852.pem
```

или

```
openssl dgst -sha1 -sign 1751852.pem  datafile
```

сгенерирует подпись в бинарном виде.

Для получения в BASE64 нужно віполнить дополнительные команды для обработки потока, например :

```
[root@ecgtest test]# cat datafile | openssl dgst -sha1 -sign 1751852.pem | uuencode -m AAA | tail -
4 | head -3
U8jjhDRYamOxeIpGqvCH3IIItTFfSyegAfQcqdwLhzyKHevpsxMV3l1RTwYw
0V7X5HK7+lMaC08JE8AEogPkbNS/JiZu1yCL47jOgJUrfUjbu9Gbob/FiiB9
8M4RI3LHOUOoeFl1uxXkEsi9fAfdtUN+mXRDI8muKT+Xu2JP6wk=
[root@ecgtest test]#
```

- ✓ Проверка ответа публичным ключом шлюза

Для начала нужно извлечь из сертификата шлюза его публичный ключ, например :

```
[root@ecgtest test]# openssl x509 -in gateway.crt -noout -pubkey | tee gateway.pub
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDC73xJq+BISfysLKT0Pc872Agm
irgNYWUYSnQd3gihhtF3ZrBmESqpoxebTGZX0iVXS4Ulb0NAGAIge6XpoIp6GQ3X
4rtFwCZTwFOSPtXhKA3dZBECI4UuoTY/Wvr4FXSzgOMF0Izubd4uPbE2O3gOFXha
pYoNKC8V+hN+rFDfmQIDAQAB
-----END PUBLIC KEY-----
```

Поместить значение подписи в файл **signature**, а значение данных для подписи – в файл **datafile .**

Выполнить команду проверки подписи :

```
[root@ecgtest test]# openssl dgst -sha1  -verify gateway.pub -signature signature
datafile
Verified OK
```

Пример на C :

```
#include <openssl/sha.h>
#include <openssl/rsa.h>
#include <openssl/objects.h>

#include <openssl/bio.h>
#include <openssl/ssl.h>
#include <openssl/err.h>

#include <stdio.h>
#include <stdlib.h>
```

```
int main(void)
{
  int c ;
  int i = 0 ;
  unsigned len ;
  const char *data = "123" ;

  unsigned char out[128] ;
  EVP_MD_CTX ctx ;
  EVP_PKEY *pkey ;
  BIO *in ;

  printf("Start....\n") ;

  /* Initializing OpenSSL */

  SSL_load_error_strings();
  ERR_load_BIO_strings();
  OpenSSL_add_all_algorithms();

  EVP_MD_CTX_init(&ctx) ;

  in = BIO_new_file( "1751852.pem", "r" ) ;
  pkey = PEM_read_bio_PrivateKey(in, NULL, NULL, NULL) ;
  BIO_free(in) ;

  c = EVP_SignInit_ex( &ctx, EVP_sha1(), NULL ) ;
  if ( ! c)
    printf("ccode : %d\n", c) ;

  c = EVP_SignUpdate( &ctx, "123", 3) ;
  if ( ! c)
    printf("ccode : %d\n", c) ;

  c = EVP_SignFinal( &ctx, &out, &len, pkey) ;
  if ( ! c)
    printf("ccode : %d\n", c) ;

  printf("signature len=%d\n", len) ;


  for ( i=0; i<len; i++) {
    printf( "%02x ", out[i]) ;
  }

  EVP_MD_CTX_cleanup(&ctx) ;


}
```

Пример на JAVA :

1. Перед выполнение необходимо получить ключи в DER формате.

Для этого выполняем

```
openssl rsa -in 1752123.pub -inform PEM -pubin -outform DER -out 1752123public.der
openssl pkcs8 -nocrypt -in 1752123.pem -inform PEM -topk8 -outform DER -out
1752123private.der
```

2. Код на Java выглядит следующим образом

```
package com.cs.eccg.testcase;

import java.io.*;
import java.security.*;
import java.security.interfaces.RSAPrivateKey;
import java.security.interfaces.RSAPublicKey;
import java.security.spec.PKCS8EncodedKeySpec;
import java.security.spec.X509EncodedKeySpec;

mport javax.crypto.Cipher;

public class testClass {

public testClass() {
super();
}

public void doTest() {
try {

final File dataFile = new File("D:\\UPC\\batch\\datafile"); // datafile, содержит
строку параметров подписи (тот же, что и в примере из batch.rar,
который Вы высылали ранее)
final File pubKeyFile = new File("D:\\UPC\\batch\\1752123public.der"); // public key
final File privKeyFile = new File("D:\\UPC\\batch\\1752123private.der"); // private key
```

```
final byte[] data = ((new String(readFile(dataFile))).trim()).getBytes();
final byte[] pubKeyBytes = readFile(pubKeyFile);
final byte[] privKeyBytes = readFile(privKeyFile);
final KeyFactory keyFactory = KeyFactory.getInstance("RSA");

// загрузка приватного ключа
final PKCS8EncodedKeySpec privSpec = new PKCS8EncodedKeySpec(privKeyBytes);
final PrivateKey pk = (RSAPrivateKey) keyFactory.generatePrivate(privSpec);

// загрузка публичного ключа
final X509EncodedKeySpec pubSpec = new X509EncodedKeySpec(pubKeyBytes);
final PublicKey k = (RSAPublicKey) keyFactory.generatePublic(pubSpec);

// формирование подписи по данным из datafile
final Signature sg = Signature.getInstance("SHA1withRSA");
sg.initSign(pk);
sg.update(data);
final byte[] bDS = sg.sign();
final String signature = new String(Base64.encode(bDS)); // ЭТО ЗНАЧЕНИЕ
ПОДСТАВЛЯЕТСЯ В ПОЛЕ 'Signature' СТРАНИЦЫ
System.out.println("DS(ENCODED) : " + signature);

// Для проверки подписи публичным ключем можно выполнить...
Signature sg2 = Signature.getInstance("SHA1withRSA");
sg2.initVerify(k);  sg2.update(data);
System.out.println(sg2.verify(bDS));

} catch (Exception ex) {
ex.printStackTrace();
}
}

public byte[] readFile(final File file) throws FileNotFoundException, IOException {
DataInputStream dis = null;
try {
dis = new DataInputStream(new FileInputStream(file));
final byte[] data = new byte[(int) file.length()];
dis.readFully(data);
return data;
```

```
} finally {
if (dis != null) {
dis.close();
}
}
}

public static void main(String[] args) {
testClass testClass = new testClass();
testClass.doTest();
}
}
```

ASP.NET :

```
 public string CreateSignature(string signatureString)
 {
SHA1CryptoServiceProvider sha1 = new SHA1CryptoServiceProvider();
X509Certificate2 cert = new X509Certificate2(MapPath("~/App_Data/P12File.p12"), "password");
RSACryptoServiceProvider rsaCryptoIPT = (RSACryptoServiceProvider)cert.PrivateKey;
ASCIIEncoding encoder = new ASCIIEncoding();
byte[] binData = encoder.GetBytes(signatureString);
byte[] binSignature = rsaCryptoIPT.SignData(binData, sha1);
return Convert.ToBase64String(binSignature);
}
```

7. Предавторизация / Поставторизация .

Магазин может использовать вид оплаты , называемый «Предавторизация». Для этого в запросе на шлюз передается дополнительный параметр с именем **Delay .**

Параметр должен иметь значение, равное «1». Соответственно, при наложении и проверке подписи он указывается рядос с полем OrderID, через запятую.

Данный вид оплаты может применяться в такой деятельности, где сначала бронируется сумма на карточке, но в расчеты данная транзакция может идти с другой суммой. Например, в гостиничной бизнесе при предварительной оплате номера и т.д.

Последовательность такая –

1. Магазин шлет запрос с параметром Delay=1 и нужной суммой
2. Кардхолдер проходит обычную процедуру оплаты используя схему 3D-Secure или ввод кода CVC2.
3. При успешной транзакции происходит блокировка средств кардхолдера, транзакции присваивается тип операции «Преавторизация» .
4. Далее, в расчеты эта транзакция не идет. Для оплаты (перевода на счет магазина), администратор магазина в интерфейсе должен ее выбрать (Поиск : Тип операции=Преавторизация, Код транзакции=Успешно) и указать окончательную сумму для оплаты.
5. На данный момент эта операция имеет следующие ограничения –
   - окончательная сумма платежа не может превышать 20% от суммы оригинальной транзакции
   - после 30 дней транзакции «Преавторизация» автоматически удаляются
6. После успешной окончательной оплаты транзакция «Преавторизация» превращается в тип «Пост-авторизация», что говорит о факте совершения окончательной оплаты и формируется новая транзакция для расчетов и типом «Покупка».
7. Дополнительно, на успешную транзакция «Покупка» может быть выполнена только одна операция «Возврат» .

Возможные коды ошибок

506 - Срок оплаты транзакции «Преавторизация» истек . (более 30 дней)
507 – Оплата транзакции «Преавторизация» уже была произведена (поа\вторная попытка)
508 – Неправильная сумма для оплаты (неправильное значение или значение больше на 20% от суммы оригинальной транзакции)