

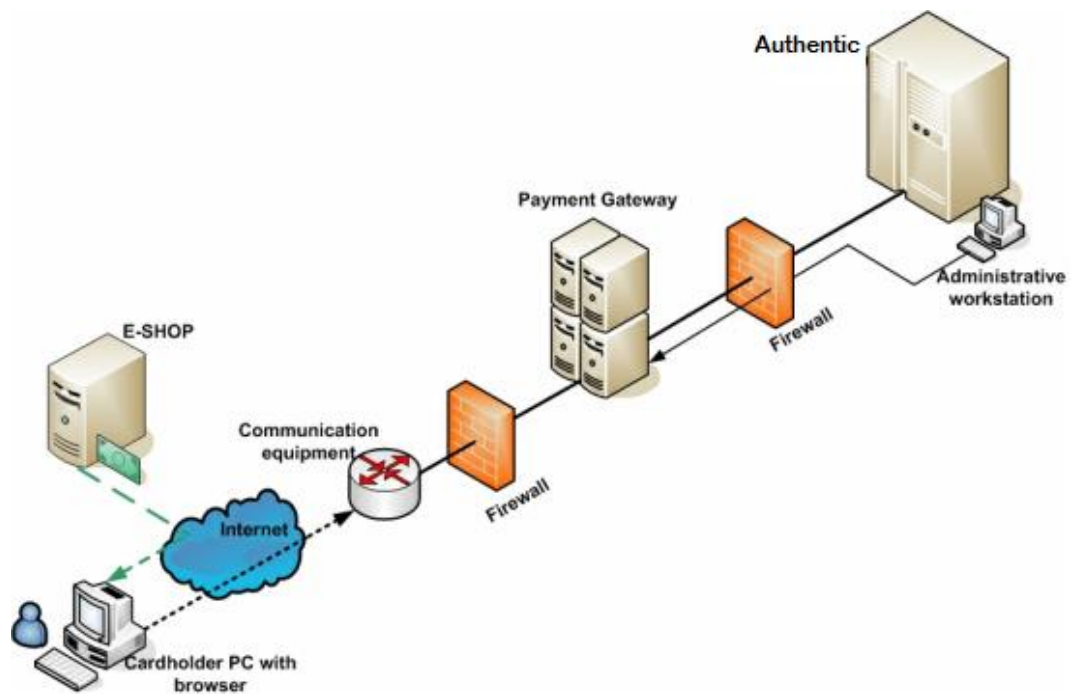


**Payment Gateway “e- Commerce Connect Gateway”
Communication Interface
E-shop administrator Guide**

1.	General positions	3
2.	Step-by Step Guide for Merchants.....	4
3.	Key Generation	6
4.	Data transmitted by the Merchant.....	9
5.	Back-off of the authorization request processing results to the e-shop	11
6.	Request for refund/reversal from the Merchant.....	13
7.	NOTIFY_URL.....	15
8.	Transaction response codes	18
9.	Request of transaction status on the Merchant side	19
10.	Preauthorisation / Postauthorisation	19
11.	Example of the programs.....	20
12.	Tokenization service	21
13.	Examples json object	24

1. General positions

During the phase of a card's purchasing capacity verification, an interaction of an E-shop with a payment gateway is performed at the end of a so-called "checkout" process. For this phase, as a general rule, it is typical that a customer has already identified a list of purchases and services, their costs, delivery terms etc. and has agreed to make a payment with a credit card. At this moment, the main task of the E-shop is to redirect the customer to a secure page of a payment server as well as to transmit all necessary transaction data via a redirection line.



After the redirection to the gateway's secure page is completed, interaction with the customer is implemented through a secure https protocol. For this purpose, the payment gateway is provided with a SSL-certificate issued by a certified agency (for example, the "VeriSign" agency). However, for the purpose of shop authentication and in order to protect data from modifications during the redirection process, all critical data is protected using MAC (Message Authentication Code).

For the interaction with the gateway, E-shop's software has to have the following pages:

1. Page with prepared values for a request transfer to the payment gateway.
2. Page (**SUCCESS_URL**) for redirection of user's browser in case of a successful transaction. Processing results are transmitted in response parameters.
3. Page (**FAILURE_URL**) for redirection of user's browser in case of an unsuccessful transaction. Processing results are transmitted in response parameters.
4. Page (**NOTIFY_URL**) for a transfer of transaction results from the gateway directly to the E-shop (optional).

If page 4 is not used, all processing results are transferred through the browser's page to the E-shop's address (pages 2, 3). Deployment of this page makes it possible to transfer transaction results directly to the Merchant from the gateway. Thereby, it allows raising a security level – the Merchant relies on the connection from the gateway's side (the address of such source is fixed), as opposed to the customer's browser. In addition, after such approval, during the customer redirection (p. 2, 3), in the response parameters, only uncritical portion of processing results is transferred, thus ensuring the concealment of the most critical data from the customer.

Some software uses dynamic elements for URL formation. Usually, this happens when either server software or a browser do not support or switch off cookies support system. In this case, the Merchant should provide a URL formation scheme.

2.Step-by Step Guide for Merchants

The Merchant (the owner of the E-shop) downloads and fills out a registration form. It can be found at <http://ecommerce.upc.ua/site/docs.html>. Forms once filled out shall be sent to ec@upc.ua.

After some time, an e-mail of the following content containing Merchant's login and password will be received from UPC:

Good time of the day,

Data for testing:

EUTELNET.COM

MerchantID= 1755637

TerminalID= E7883657

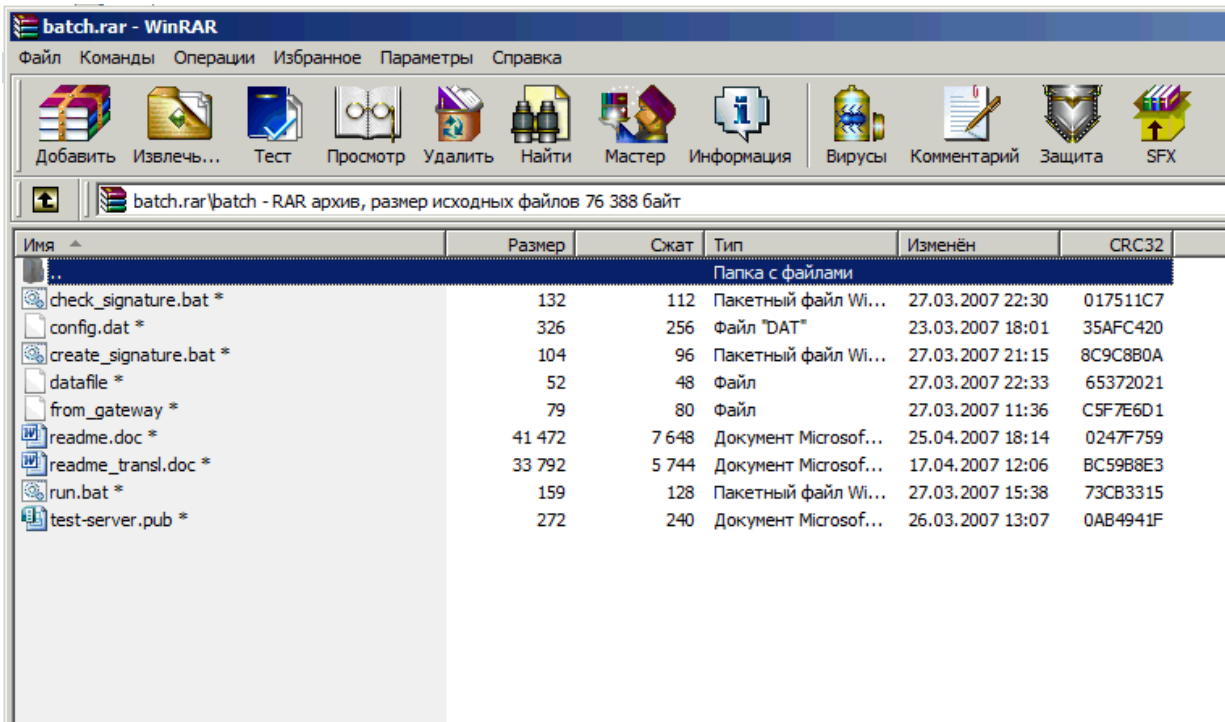
Merchant interface: <https://ecg.test.upc.ua/go/merchant/> Login/Password: 1755637/1755637

Gateway address: <https://ecg.test.upc.ua/go/enter>

Sever certificate: In the attachment. All necessary documentation can be found here: (See attached file: shop_gateway_interface_eng.doc)

It is necessary for you to send us a Merchant certificate (file named 1755637.crt)

In the batch.7z archive, please find documents and examples of key and signature generations.



The Merchant shall follow the link <https://ecg.test.upc.ua/go/merchant/> and change the password on the profile page:

2020-02-24 11:20:45 EET

MENU

- Info
- Profile**
- Terminals
- Transactions
- Error codes
- Stop List
- Invoicing

Profile

Login: igor

First name: not entered yet

Last name: not entered yet

City: Kyiv

E-mail: not entered yet@

Phone number:

Address:

Update Reset

Change password

Current password:

Password:

Re:Password:

Update Reset

On the "Terminal" page, the Merchant shall select website and indicate the URL for the pages with a successful and unsuccessful transactions:

January 23, 2013 3:36:47 PM

MENU

- Info
- Profile
- Terminals**
- Transactions
- Error codes
- Stop List
- Invoicing

Terminal data

Merchant ID: 1752845

Terminal ID: E7880845

Merchant: hbs-ukraine.com

Settlement time: -AUTO -AUTO (hh:mm)

Number of attempts to enter card: 5

Success URL (SUCCESS_URL): http://b2e-serg.611.qa.loc/UI_NET/Booking/PaymentResponse.aspx

Failure URL (FAILURE_URL): http://b2e-serg.611.qa.loc/UI_NET/Booking/PaymentResponse.aspx

Notify URL (NOTIFY_URL):

Revers transaction on unsuccessful notification (NOTIFY_URL): No

E-Mail: volkov@elegant-travel.com.ua

To accept cards put into Stop-list by other merchants: Yes

Update Reset

It is also necessary to generate and send us a certificate (it should contain public and private keys)

3.Key Generation

OpenSSL Setup

Prior to proceeding, it is necessary to download Win32 OpenSSL which can be found at:
<https://www.openssl.org/source/>

After the software is set up, it is important to enter a variable Path for a bin catalogue.

1. In system variables:

Variable PATH → "Change"

Put a semicolon at the end of the line and enter a pathway to the bin folder: c:\OpenSSL\bin

For the next step, in the console, open the batch folder and run run.bat (one of the attachments in the e-mail received from UPC)

2. The command can be performed for example in FAR. For this, place the cursor on run.bat file, press Ctrl Enter and add MERCHANT ID with a space

Key Generation

Key generation and exchange is performed after the Merchant sends a request for registration and receives E-shop attributes via Internet (that includes a Merchant ID)

Prior to key generation, it is necessary to edit the file config.dat according to the data of the E-shop. Config.dat data shall not strictly correspond with the data in the request; such data is used for neither signature generation nor signature verification and is only used for an identification of a certificate file.

```
[ req ]
prompt                = no
distinguished_name    = req_distinguished_name

[ req_distinguished_name ]
#Страна
CN=UA

#Область
ST=Kievskaya

#Город
L=Kiev

#Название организации
O=upc.ua

#Название отделения
OU=ECOMMERCE

#Имя для сертификата (Ваше имя)
CN=SERGEI

#Email организации
emailAddress=S.Sichnoy@upc.ua
```

Run.bat command with the MerchantID parameter (e.g. run.bat 1770000) generates three files:

1770000.pem – private key
 1770000.pub – public key
 1770000.crt – certificate

Now, it is necessary to send the file 1753019.crt to ec@upc.ua and wait for a response from UPC. As a response, we send an answer informing you that the certificate has been loaded.

On a server, the Merchant is required to have at least the following:

1. test-server.cert – file sent by UPC in the registration confirmation e-mail (is used for a response verification)
2. pem – file (private key) generated by the Merchant (is used for the signature of delivered data)

Signature Generation

Signature is generated based on two files: *.pem and datafile. Datafile contains data (fields), for which the signature is generated.

Note that the fields sequence should be kept, otherwise the request will be rejected with 405 code (Signature is invalid).

The fields are recorded in datafile in the following sequence (this sequence should also be followed for software implementation):

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

The number of ; signs should remain the same. If a field is missing, ;; should be used. For example, SessionData(SD) field is missing, thus datafile will be as follows:

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

In case of Delay, AltCurrency or AltAmount fields are missing, comma is omitted before these fields. For example:

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

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

○ *MerchantId;TerminalId;PurchaseTime;OrderId;CurrencyId;Amount;;*

For a correct signature generation, the datafile should not contain any extra symbols (spaces, returns, line breaks). The datafile has to be checked for unnecessary symbols in a HEX or FAR editor (F3, F4).

The data should be arranged likewise in case of a software-based signature processing.

For the signature generation, create_signature.bat should be launched with a *.pem value, e.g., create_signature.bat 1770000.pem. As a result, two files will be updated or created: signature.bin (a signature) and signature (a base64-encoded signature). Signature file data is sent in the request as a signature. (Note: in the request, you cannot enter a field name in the lower case, i.e., field named "merchantid" will not be registered as correct).

Note: Signature generation (code)

Please see examples at the end of the document.

If the transaction is successful, the script will return "good" which means that the payment was successful.

Note:

It is important to keep a \$data variable correct. It is generated the following way:

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

The rules are the same as for Delay, AltCurrencyId, AltAmount fields – the comma is deleted in front of them. If in the request to the gateway is present field Ref3, it shall be included in signature creation. For example:

MerchantId;TerminalId;PurchaseTime;OrderId;CurrencyId;Amount;SessionData(SD);Ref3;

Request for refund/reversal (signature generation)

If there is a request to refund/reversal the data shall be added with the following fields: ApprovalCode, RRN. In case when the request contains fields RefundAmount and/or Ref3, these fields shall be included in the signature, please see an example datafile using all fields:

*MerchantId;TerminalId;PurchaseTime;OrderId;CurrencyId;Amount;SessionData(SD);ApprovalCode
;RRN;RefundAmount;Ref3;*

If the optional field (RefundAmount or Ref3) is missing, this field shall be not included. Example:

*MerchantId;TerminalId;PurchaseTime;OrderId;CurrencyId;Amount;SessionData(SD);ApprovalCode
;RRN;Ref3;*

*MerchantId;TerminalId;PurchaseTime;OrderId;CurrencyId;Amount;SessionData(SD);ApprovalCode
;RRN;RefundAmount;*

*MerchantId;TerminalId;PurchaseTime;OrderId;CurrencyId;Amount;SessionData(SD);ApprovalCode
;RRN;*

For correct signature creation datafile shall not contain extra characters (spaces, new line characters, characters of return to the beginning of the line). Shall be checked that there are no extra symbols in HEX sditor or in FARE (F3, F4). In the same way this data shall be present during program implementation of the signature.

To create the signature please un create_signature.bat with parameter *.pem.

For example, create_signature.bat 1770000.pem.

As a result two files will be updated or created: signature.bin (signature) and signature (signature in code base64). Data in file signature shall be sent in request as a signature. (Important. In the request the name of the fields cannot be input in lowercase, so field with name merchantid is not considered as one).

Signature Verification

The fields should be entered into the from_gateway file for gateway data signature verification in the following order:

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

All requirements from the previous item are also obligatory for from_gateway generation.

Gateway data signature should be placed into a signature file for verification.

Note that the number of significant symbols in one line of the signature file should not exceed 64 (the length of the line).

Launch check_signature.bat to verify the signature.

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

4.Data transmitted by the Merchant

The E-shop has to transfer a number of parameters when passing to a secure page of the gateway. Such parameters are indicated in the following Table 1:

Table 1

Parameter	Structure	Format	Description of the parameter	Additional comments
Version	F	n4	Version of the interface SG	Version of the interface protocol. Current version is 0001. This is a help parameter for the handler of the gateway incoming data. It is used to choose a better option for data processing.
MerchantID	L	an15	Merchant identifier	Assigned by processing bank.
TerminalID	F	an8	Terminal identifier	-- // --
TotalAmount	F	n12	Purchase amount	In the smallest currency units (kopecks, cents)
Currency	F	n3	Currency	Under the agreement with the processing bank.
AltCurrency	F	n3	Alternative currency	Optional parameter Is used in case the shop wants to indicate the payment amount in a different currency.
AltTotalAmount (O)	F	N1..12	Order amount (alternative currency)	Optional parameter In the smallest currency units (kopecks, cents)

PurchaseTime	F	n12	Time of the purchase in MMddhhmmss format	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 зони - [+ -] Hours Minutes Example +0300 , -0200 If a zone is not indicated in the parameters, it is considered to be same as the gateway's
Locale	F	a2	Language of the interface (en, ru, uk)	Language of the interface of the secure gateway page.
OrderID	L	ans...20	Number of the order up to 20 byte in length	The value of the XID is determined based on the OrderID. If the OrderID can not be used, one should use the XID parameter.
SD (O)	Var	an...99	Session Data – session's data	Auxiliary parameter which can be used by the e-shop in order to administer users' sessions.
PurchaseDesc (O)	L	ans...125	Brief description of the purchase	Optional parameter stipulated by the 3-D Secure specification.
Signature	Var	an...40	MAC-code value	The length of the parameter depends on the chosen scheme of MAC-code calculation.
Delay	F	N1	Preauthorization payment identifier	For preauthorization, the value should be equal to "1" otherwise 0 or empty
Ref3 (O)	L	Ans 1 150		

Annotation:*A. Structure description*

F – full field
L – left justified
R – right justified
S – filled with spaces
Z – filled with zeroes
Var –variable length field

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,

Note: AltTotalAmount, AltCurrency parameters are defined if the merchant needs to indicate the payment amount in a currency which is different from the currency in the agreement with the acquiring bank. At the same time, 4 parameters have to be sent to the gateway:

TotalAmount, Currency – amount and currency according to the terms and conditions of the agreement. Please keep in mind that a transaction will be performed according to the TotalAmount, Currency parameters. The Merchant is responsible for matching the total amount between two different currencies (correct calculations according to the exchange rate). AltTotalAmount, AltCurrency – total amount and currency indicated for a payment at the shop. Currency codes: 643 Russian Ruble, 840 United States Dollar, 978 Euro, 980 Ukrainian Hryvnia

These parameters are transferred to the gateway's page in a certain HTML-format using the HTTPS/POST method for a further input of the payment card details by the customer (cardholder).

Example:

```
<form method="POST" action="https://ecg.test.upc.ua/go/enter">
<input type="hidden" value="1" name="Version">
<input type="hidden" value="1700000" name="MerchantID">
<input type="hidden" value="E7000000" name="TerminalID">
<input type="hidden" value="30000" name="TotalAmount">
<input type="hidden" value="980" name="Currency">
<input type="hidden" value="ru" name="locale">
<input type="hidden" value="0001" name="OrderID">
<input type="hidden" name="SD" value="sdfhsdfsdfn3432n4jn23j4">
<input type="hidden" value="131222155090" name="PurchaseTime">
<input type="hidden" value="tran_test" name="PurchaseDesc">
<input type="hidden" name="Signature" value=".....">
<input type="submit">
</form>
```

Later on, at the gateway's page, received data is supplemented with the Card Number, ExpYear, ExpMonth, CVV2, and Card Type. Prior to that, the gateway performs a sequence of verifications (the existence of registration parameters of the Merchant in the database, correspondence of the currency to a registered value, authorization limit of the Merchant, verification of the electronic signature).

After that, the gateway provides the customer's browser with the page to input the payment card details. At the same time, the buyer can indicate the card type (with a condition that the Merchant accepts the card type). Also, the customer can input a CVV2 code (for MEASTRO cards, this function is not supported).

At the next stage, the request processing is performed using either a 3D-Secure or standard scheme (Ecommerce channel encryption), depending on the parameters of the bank that provides services.

5.Back-off of the authorization request processing results to the e-shop

Processing results (transaction results) can be transferred in two ways:

- forwarding of the results to NOTIFY_URL address and redirection of the customer's browser to the page "successful/ unsuccessful"
- forwarding of the results through the customer's browser to the page "successful/unsuccessful"

In the first case, the processing results are transferred from the gateway to the E-shop's page using the HTTP/HTTPS POST method. Under such conditions, a level of additional security can be achieved for the E-shop through limiting access to a particular URL to the gateway requests only.

The gateway at the session might receive a confirmation of the E-shop notification concerning the state and parameters of a performed transaction. One of the advantages is that no parameters of the reverse transaction will be at the customer's browser page

A list of response parameters to the E-shop website (see Table 2)

Table 2

Parameter	Format	Description of the parameter	Additional comments
MerchantID	an15	Merchant identifier	Is similar to the data in the authorization request
TerminalID	an8	Terminal identifier	
TotalAmount	n12	Purchase amount	
Currency	n3	Currency	
PurchaseTime	n12	Time of the purchase request (YYMMDDhhmmss)	
OrderID	ans..20	Order ID	
XID	ans28	Transaction identifier (number of the order augmented up to 20 byte)	
SD	an... 99	Session Data	
ApprovalCode	n6	Host authorization code	
Rrn	n10	Retrieval Reference Number	Unique transaction number in the authorization and settlement system of the servicing bank
ProxyPan	N13...19	Lust 4 digits of the card number	PAN value (four last digits) with the additional zeroes in front for the PAN length
TranCode	n3	Code of the transaction completion	
Signature	an...40	MAC-code value for the chosen scheme of the gateway/e-shop intercommunication	Parameter length depends on the chosen scheme of the MAC-code calculation

After the given session of the gateway with the E-shop host is finished, a final forwarding of the browser takes place. It looks like "approved"/"rejected" transaction. A minimum number of parameters are transferred such as: OrderID , TranCode and SD.

Addresses of the E-shop's Web Pages retrieved by the gateway from its Data Base, i.e. have to be provided by the Merchant beforehand – at the stage of registration.

In the second case, the processing results are transferred through the browser's page, where a corresponding form is transmitted to the Merchant's website address to the page "successful/unsuccessful". The

operation of the form initiation is performed by Java Script. If implementation of this language is not possible, the message about a necessity to manually confirm a form of sending is displayed.

For connecting the customer with a corresponding E-shop's session, the purchase SD (Session Data) parameter is used, which is transferred through the customer's browser in the process of a backward redirection.

Example:

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

6. Request for refund/reversal from the Merchant

Request for refund/reversal shall be performed for authorization transaction only. To perform reversal the shop shall send the request to the gateway. It can be performed via sending POST request to the gateway page with parameters states in Table 3:

Table 3

Parameter	Structure	Format	Parameter name (purpose)	Additional comment
MerchantID	L	an15	Merchant ID	Assigned by the bank acquirer
TerminalID	F	an8	Terminal ID	-- // --
TotalAmount	F	N1..12	Amount of the request	Shall be indicated in small currency units (cents)
Currency	F	n3	Currency	Defined in agreement with bank acquirer
PurchaseTime	F	n12..17	Time of the request in format yyMMddHHmmss or yyMMddHHmmssZ	
OrderID	L	Ans...20	Order number with length up to 20 bytes	
ApprovalCode	F	An6	Host authorization code	
Rrn	F	n12	Retrieval Reference Number	Unique transaction number in authorization system and settlement of the bank acquirer
SD (O)	Var	an...99	Session Data – Session data	An auxiliary parameter, that can be used by the trade system to manage user sessions

Parameter	Structure	Format	Parameter name (purpose)	Additional comment
Signature	Var	Depend on the scheme	MAC-code or signature meaning	Length of the parameter depends on the selected calculation scheme
Ref3 (O)	L	ans..1 150		

Gateway creates reply as a text page with parameters (see Table 4):

Table 4

Parameter	Format	Parameter name (purpose)	Additional comment
TranCode	N3	Transaction completion code	Please refer to table 6
MerchantID	an15	MerchantID	
TerminalID	an8	Идентификатор терминала	--- // ---
CardType	an4	TerminalID	VISA – Visa MAST – MasterCard MAES - Maestro
ERROR (O)	ans	Contains brief information about error	Is optional and is formed only when error occurs during request processing

The reversal is considered as successful if TranCode field meaning = "000"

Example:

```
<html>
<body>
<form method='POST' action="https://ecg.test.upc.ua/go/repayment">
<input type='hidden' name='MerchantID' value='1752493' />
<input type='hidden' name='TerminalID' value=' E7880293' />
<input type='hidden' name='OrderID' value='PAY160601124534' />
<input type='hidden' name='Currency' value='980' />
<input type='hidden' name='TotalAmount' value='12550' />
<input type='hidden' name='PurchaseTime' value='160601124534' />
<input type='hidden' name='ApprovalCode' value='123456' />
<input type='hidden' name='RRN' value='222222222' />
<input type='hidden' name='RefundAmount' value='12000' />
<input type='hidden' name='Signature' value='45F345Fafde4455445Gvb550' />
<input type='submit' value='go'>
</form>
</body> <html>
```

7. NOTIFY_URL

A successful and guaranteed redirection of the browser with payment results parameters is a necessary condition for receiving the payment at the e-shop. In some cases, however, it can fail due to the following situations:

- 1) browser failure, computer freezing;
- 2) inadequate user actions at the time of a response delivery;
- 3) loss of connection with an Internet provider
- 4) incorrect work of a browser with setup security parameters which can effect browser's performance.

There can be a situation when the card payment was made but the result was not delivered to the shop. At the same time, an investigation between a purchaser and shop is initiated in order to eliminate the payment amount blocking and either set up the transaction as "Paid" or perform a return. In such cases, it is recommended for a shop to implement a scheme with a response delivery from the gateway.

The payment gateway initiates a message delivery. For the testing server, the message shall be sent from the IP address - 195.85.198.16, for production - 195.85.198.15. The results are forwarded using HTTP/HTTPS POST from the gateway to the E-shop's page (80/443 ports).

```
Notify request message:
PurchaseTime = '090929152500'
ProxyPan = '499999*****0011'
Currency = '980'
ApprovalCode = '111111'
MerchantID = '1752493'
OrderID = '11111111111111111111'
Signature = test'
Rrn = '2222222222'
XID = '333333-44444444'
Email = 's.sich@upc.ua'
SD = '24ee6084a5343e3d'
TranCode = '000'
TerminalID = 'E7880293'
TotalAmount = '500'
```

The shop returns an answer in the body of the processed page. Each parameter and its setting as Parameter=Setting shall be returned to a new line. Lines are separated with a line separation digit.

In the answer, additionally to the originally set parameters (TerminalID, OrderID, Currency, TotalAmount, XID, PurchaseTime), 3 new parameters are returned (see Table 5):

Table 5

Parameter	Setting	Description
Response.action	approve / reverse	Setting 'approve' means that a shop approves the purchase from its side Setting 'reverse' means that the gateway performs a rollback of a successful transaction and sets the 503 code of completion – «Transaction cancelled by the E-shop »
Response.reason	An ... 255	Explanation of the shop's response (optional), For example – a reason for setting <u>Response.action</u> .

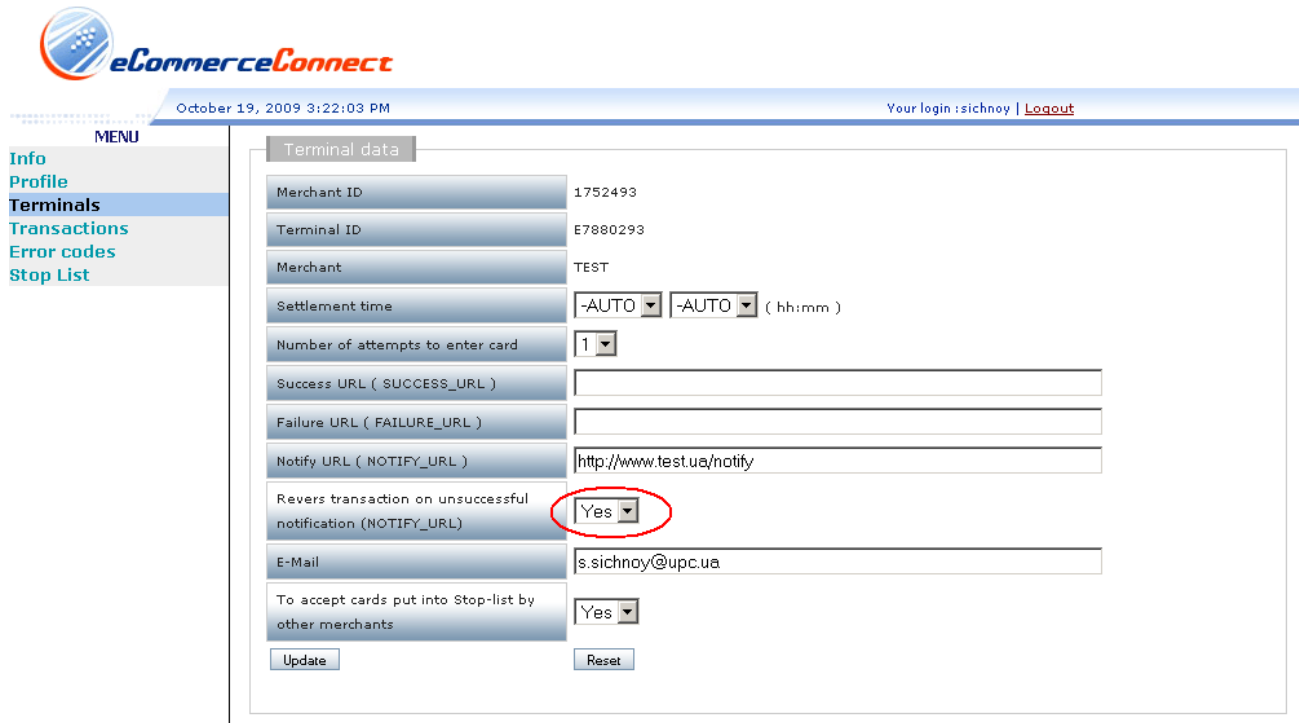
		Creation of such field is performed on the shop's side and shall be used only for shop's needs. At UPC, it is recorded in transaction logs only
Response.forwardUrl	An ... 255	Setting URL to redirect the user's browser instead of SUCCESS_URL or NOTIFY_URL. Allows using dynamic links for a client's redirect.

```

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";

```

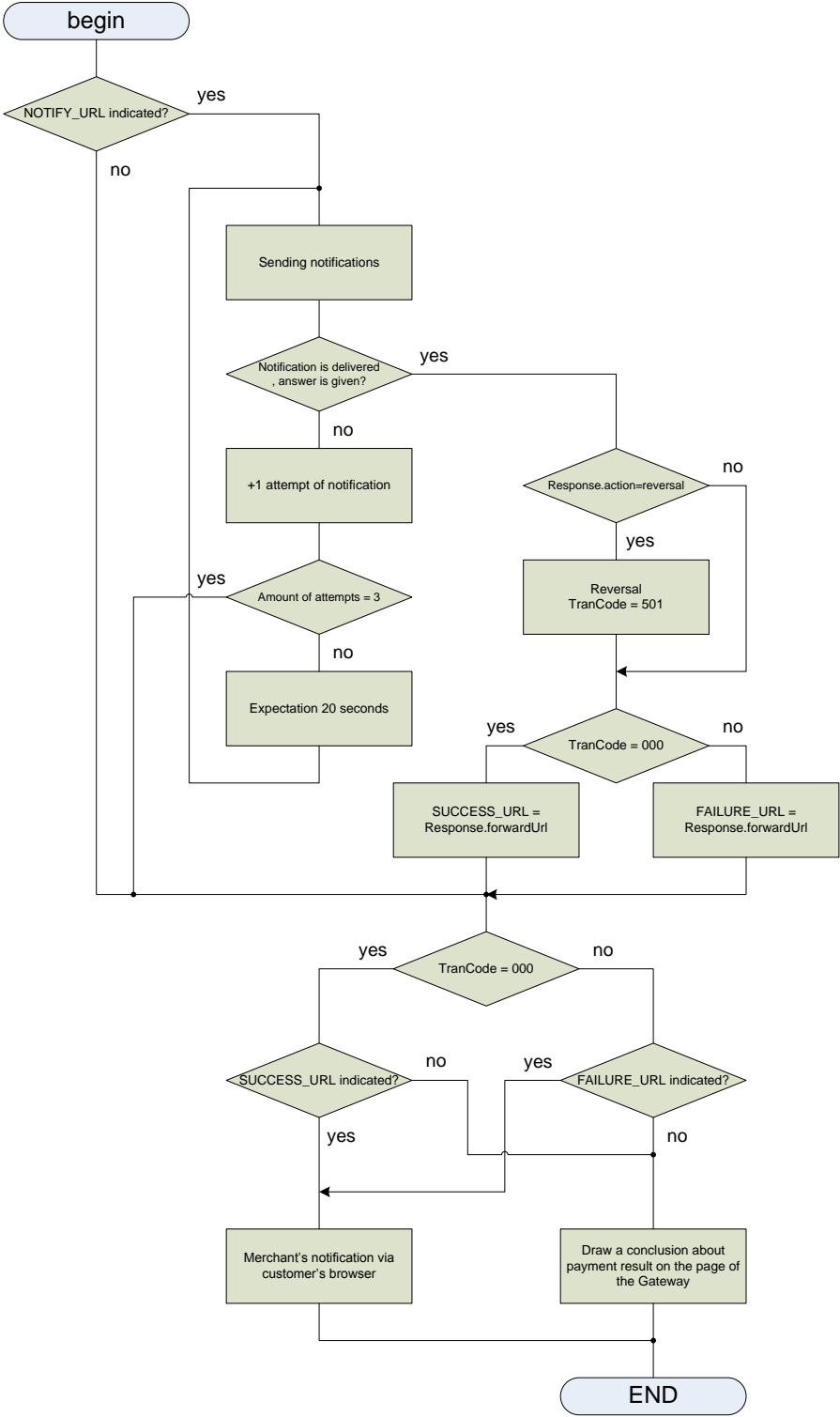
Also, a shop administrator can allow choosing or restricting transactions on which no delivery to NOTIFY_URL is performed. This scheme allows an automatic accounting of systems that provide services. It uses information from the payment gateway to avoid a discrepancy of data in case the message is not delivered to NOTIFY_URL.



The screenshot shows the eCommerceConnect web interface. The top navigation bar includes the logo, the date and time (October 19, 2009 3:22:03 PM), and the user's login information (Your login : sichnoy | Logout). A left-hand menu lists various options: Info, Profile, Terminals, Transactions, Error codes, and Stop List. The main content area is titled "Terminal data" and contains a form with the following fields:

- Merchant ID: 1752493
- Terminal ID: E7880293
- Merchant: TEST
- Settlement time: -AUTO (dropdown) -AUTO (dropdown) (hh:mm)
- Number of attempts to enter card: 1 (dropdown)
- Success URL (SUCCESS_URL): [empty text box]
- Failure URL (FAILURE_URL): [empty text box]
- Notify URL (NOTIFY_URL): http://www.test.ua/notify
- Revers transaction on unsuccessful notification (NOTIFY_URL): Yes (dropdown, circled in red)
- E-Mail: s.sichnoy@upc.ua
- To accept cards put into Stop-list by other merchants: Yes (dropdown)

At the bottom of the form, there are "Update" and "Reset" buttons.



Such method of transferring the answer to the shop is preferable and recommended. It allows decreasing the number of incorrectly completed transaction (for example, caused by errors of user's browsers or incorrect actions). Even in case of problem situations, the E-shop will have reliable information on transaction processing results.

In such case, if the original transaction has an operation type “purchase” and a transaction code “504/This request for payment is not permitted by the gateway”, a rollback is performed automatically by the payment server with a transaction code “000/Transaction is completed successfully”

Below is the logic of the answer delivery from the gateway.

The screenshot shows the eCommerceConnect interface. At the top, it displays the date and time (October 19, 2009 3:25:02 PM) and the user's login information (Your login: zsihney | Logout). A menu on the left includes options like Info, Profile, Terminals, Transactions, Error codes, and Stop List. The main content area shows a search results page for transactions. A search form at the top allows filtering by Merchant (TEST), Order ID (2222245224), Operation Type, Batch, Approval code, and Transaction code. Below the search form is a table with the following data:

ID	Merchant	ID-addr	Order ID	Operation Type	Description	Time	Currency	Amount	Approval code	Transaction code	Batch
18590	TEST	195.85.198.51	2222245224	Reverzal	Tarns_2	Apr 23, 2009 11:56:34 AM	980	500	-	000 / Transaction is approved	54522
18589	TEST	195.85.198.51	2222245224	Purchase	Tarns_2	Apr 23, 2009 11:56:32 AM	980	500	397985	504 / The payment transaction was canceled by gateway	54522

Below the table, it indicates 'Pages 1 (2)' and an 'Export' button.

8. Transaction response codes

Transaction response codes are divided into several classes and subclasses and are used to inform the Merchant about transaction results. To indicate a successful transaction, one response code is required. A major part of response codes provides generalized information about the reasons for an unsuccessful transaction to the Merchant. (see Table 6)

Table 6

Codes on basis of the authorization host responses		Comments
Integrated response codes for the e-shops	Interpretation of the codes	Response codes in the message 1110
000	Successful authorization	00x
105	Do not honor by the issuing bank	100, 103,104,105...107,
116	Insufficient funds	116
111	Non-existent card	111,125,200,202
108	Lost or stolen card	208,209
101	Invalid expiration date	101,201
130	Amount limit exceeded	121,123
290	Issuer is inaccessible	905...908,910
291	Technical or communicational problem	9xx (except indicated above)

At the end of the document you can see all the response codes. (see Table 8)

9. Request of transaction status on the Merchant side

To receive payment status for the Merchant it is recommended to use scheme including NOTIFY_URL. In this case the gateway will make an attempt to deliver the results to the shop directly, not relaying on sending parameters via user browser.

Additionally shop can send the request on transaction status from its side with the following parameters

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

The gateway returns text page with additional parameters –

XID=
TranCode=
ApprovalCode=

The transaction is considered as successful if TranCode field meaning = "000".

This mechanism of authorization results delivery is considered as optional, but can be used in case some issues occur with results delivery via cardholder browser.

Example:

```
<html>
<body>
<form method='POST' action="https://ecg.test.upc.ua/go/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>
```

10. Preauthorisation / Postauthorisation

The Merchant can use a type of payment called "Preauthorisation". A request is sent to the gateway, and it contains an additional parameter called **Delay**.

The Parameter shall bear a value of "1". It is entered next to the field OrderId along with a signature entering and verification. They are separated with a comma.

This type of payment is used when the amount is reserved on the card but another amount can be settled. For example, it can be used in the hotel business to make a prepayment for a room.

The procedure is as follows:

1. The Merchant sends a request with a parameter Delay=1 and with an amount needed.

2. Cardholder has a usual procedure of payment using a 3D-Secure schema or CVC2 entering.

3. In case of a successful transaction, cardholder's funds are blocked, and a transaction is assigned a "Preauthorisation" operation type.

4. This transaction is not taken to further settlement. For the payment (transfer to the account of the Merchant), the Merchant's administrator shall choose it (Find: Type of operation=Preauthorisation, Code of transaction=Successful) and put a final sum for payment.

5. This transaction has some restrictions:

- the final amount cannot exceed 20% of the amount of the initial transaction;
- after 30 days of a "Preauthorisation" transaction, it is deleted automatically.

6. After a successful final payment, the "Preauthorisation" transaction changes to "Postauthorisation".

This shows that the final payment is completed, and a new transaction for payment called "Purchase" is formed.

7. In addition to a successful "Purchase" transaction, only one "Return" operation can be performed.

Possible errors:

506 – Time of payment for "Preauthorisation" transaction is over. (more than 30 days)

507 – Payment for "Preauthorisation" transaction was made before (repeated attempt)

508 – Wrong amount for payment (value is wrong or 20% more than the initial transaction)

11.Example of the programs

Example in PHP :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Buy</title>
</head>
<body>

<?php
$MerchantID = '1753019';
$TerminalID = 'E7881019';
$OrderID = 19;
$PurchaseTime = date("ymdHis") ;
$TotalAmount = 242;
$CurrencyID = 980;
$data = "$MerchantID;$TerminalID;$PurchaseTime;$OrderID;$CurrencyID;$TotalAmount;";
$fp = fopen("$MerchantID.pem", "r");
$priv_key = fread($fp, 8192);
fclose($fp);
$pkid = openssl_get_privatekey($priv_key);
openssl_sign( $data* , $signature, $pkid);
openssl_free_key($pkid);
$b64sign = base64_encode($signature);
?>

<form action="https://ecg.test.upc.ua/go/enter/" method="post" >
  <input name="Version" type="hidden" value="1" />
  <input name="MerchantID" type="hidden" value="<?php echo $MerchantID?>" />
  <input name="TerminalID" type="hidden" value="<?php echo $TerminalID?>" />
  <input name="TotalAmount" type="hidden" value="<?php echo $TotalAmount?>" />
```

```

<input name="Currency" type="hidden" value="<?php echo $CurrencyID?>" />
<input name="locale" type="hidden" value="RU" />
<input name="PurchaseTime" type="hidden" value="<?php echo $PurchaseTime ?>" />
<input name="OrderID" type="hidden" value="<?php echo $OrderID?>" />
<input name="Signature" type="hidden" value="<?php echo "$b64sign" ?>" />
Sum: <?php echo $TotalAmount?> <input type="submit"/>
</form>

</body>
</html>

```

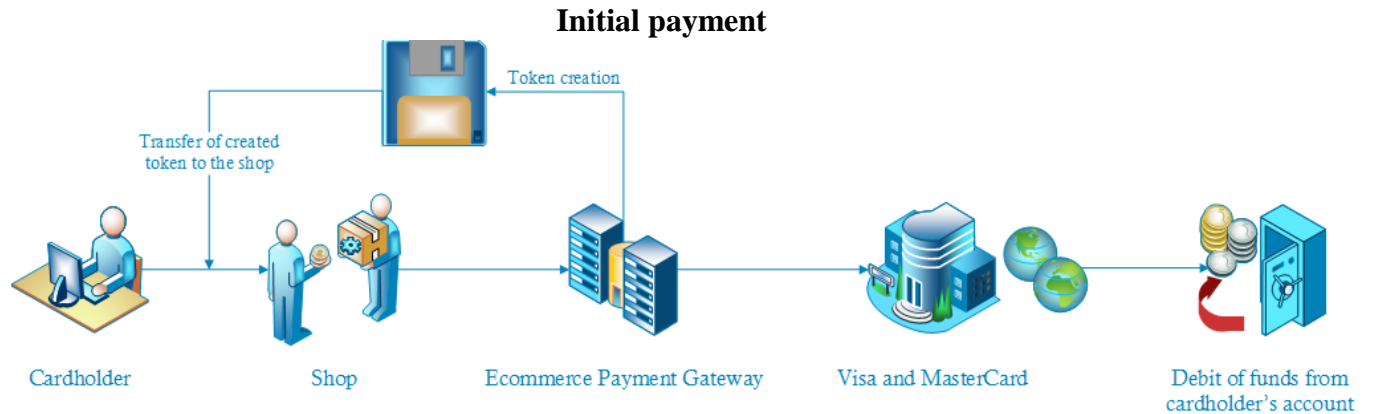
***Note**

\$data is generated the following way:

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

12.Tokenization service

The service is designed to enable Merchants to create payment card digital analogue (token) on UPC side. The service will allow Non-PCIDSS merchants to store payment card digital analogue (token) and initiate transactions using this token. Cardholder enters payment data during the first purchase. After the successful payment by the cardholder with full card requisites, the Token will be assigned to this card and will be sent to the shop in the response message.



Further payments the merchant can initiate using Token. Payment by token is a debit of funds from payment card without repetitive card requisites input.

Subsequent transactions



To activate Tokenization service, the merchant should send a request to ec@upc.ua and indicate Merchant_id. The Merchant should also notify Acquirer bank about such request.

The Acquirer bank sends the limit with the threshold amounts (concerning currencies used by the bank), in case the limit is exceeded, the request shall contain CVC (of the card for which the token has been created). The Acquirer bank can set individual limits with the threshold amounts for the merchants.

When the option is activated, the gateway creates response to the transaction with the parameters indicated in the Table 7

Table 7

Parameter	Format	Parameter name (aim)	Comment
MerchantID	an15	Merchant ID	Is the same as in the authorization request
TerminalID	an8	Terminal ID	--- // ---
TotalAmount	n..12	Order amount	--- // ---
AltTotalAmount	n..12		
Currency	n3	Currency	--- // ---
AltCurrency	n3		
PurchaseTime	N12	Time of the order (YYMMDDhhmmss)	--- // ---
OrderID	ans..20	Order ID	
XID	ans28	Transaction ID (Order ID supplemented by 20 symbols)	--- // ---
SD	an... 99	Session data	--- // ---
ApprovalCode	An6	Host authorization code	
Rrn	N12	Retrieval Reference Number	Unique transaction number in the authorization system and settlement of the servicing bank.
ProxyPan	N13...19	4 last digits of the card number	PAN value (4 last digits) with zeros at the beginning in order to indicate the PAN length.
TranCode	N3	Transaction completion code	See table. 6
Signature	An...40	MAC-code value for the selected interaction scheme gateway-shop	Parameter length depends on the selected scheme of the MAC-code calculation
Delay	N1	Payment ID Preauthorization	
UPCToken	An...32	Payment card digital analogue	Card/card expiry date/Merchant_id
UPCTokenExp		Token expiry dare, format (MMYYYY)	

Processing results are sent using HTTP/HTTPS POST method from gateway to the shop page

Notify request message:

PurchaseTime = '090929152500'
ProxyPan = '499999*****0011'
Currency = '980'
ApprovalCode = '111111'
MerchantID = '1752493'
OrderID = '11111111111111111111'
Signature = test'
Rrn = '2222222222'
XID = '333333-44444444'
Email = ec@upc.ua'
SD = '24ee6084a5343e3d'

UPCToken = '254484kC162EEC13E5B012736288683AC'
TranCode = '000'
TerminalID = 'E7880293'
TotalAmount = '500'

The merchant can use the received token for subsequent transaction completion.

Request for transaction shall be created in JSON Web Signature format (Standard JWS (rfc7515. See <https://tools.ietf.org/html/rfc7515#page-10>)). All data shall be transferred in BASE64URL coding. [json](#) object creation is described in section «[Examples](#)»

8. Examples json object

```
header => {"alg":"RS256"}
Convert in Base64URL header
eyJhbGciOiJSUzI1NiJ9
```

```
Data to create payload
{MerchantID:"77777701001",TerminalID:"E0177771",OrderID:"orderToken1",UPCToken:"1068955433FEBE8
B2F237B94A0B10ADC",TotalAmount:100,Currency:980,PurchaseTime:"180919174126",PurchaseDesc:"Test
token"}
```

```
Convert in Base64URL payload
e01lcmNoYW50SUQ6Ijc3Nzc3NzAxMDAxIixUZXRtaW5hbEIEOiJFMDE3Nzc3MSIsT3JkZXJRRDob3JkZXJU
b2tlbjEiLFVQQ1Rva2VuOiIxMDY4OTU1NDMzRkVCRThCMkYyMzdCOTRBMEIxMEFEQyIsVG90YWxBb
W91bnQ6MTAwLEN1cnJlbnN5Ojk4MCxQdXJjaGFzZVRpbWU6IjE4MDkxOTE3NDEyNiIsUHVyY2hhc2VE
ZXNjOiJUZlZXRva2VuIn0=
```

Create signature based on the received data.

prepare signature= datafile1 ="header"."payload"

```
eyJhbGciOiJSUzI1NiJ9.e01lcmNoYW50SUQ6Ijc3Nzc3NzAxMDAxIixUZXRtaW5hbEIEOiJFMDE3Nzc3MSIsT
3JkZXJRRDob3JkZXJUb2tlbjEiLFVQQ1Rva2VuOiIxMDY4OTU1NDMzRkVCRThCMkYyMzdCOTRBMEIx
MEFEQyIsVG90YWxBbW91bnQ6MTAwLEN1cnJlbnN5Ojk4MCxQdXJjaGFzZVRpbWU6IjE4MDkxOTE3N
DEyNiIsUHVyY2hhc2VEZXNjOiJUZlZXRva2VuIn0=
```

Using example: openssl

```
openssl dgst -sha256 -sign %1 datafile1 > signature.bin
```

and convert to

```
openssl base64 -e -in signature. -out signature
```

```
signature= EXDEhK9kMK0lwTEWH4mm1oJvKm5vVFyXnyDnqEDHDc3mYyXEhLv3Ih6_fdmN-
```

```
apUPxgV5GEpV0YQWTuSyGF3o32dF0n-
```

```
A4LrZ93z8Dw7gj9ULLd5ffRE42x0tFL6jNNEnVUbj8WB1UeR6mRN4l4aTRaNU123hq6UIqB_jsTxWJU
```

Create json request to <https://ecg.test.upc.ua/go/payByToken>

```
{
header:"eyJhbGciOiJSUzI1NiJ9",
payload:"e01lcmNoYW50SUQ6IjE3NTIzMzkiLFRlcm1pbmFsSUQ6IklU3ODgwNTM5IixPcmRlckIEOiJ0b2tlbjEiLFV
sVVBdVG9rZW46IjI1NDQ4NEMxNjJFRUMxM0U1QjAxMjczNjI4ODY4M0FDlixUb3RhbEFTb3VudDo1MD
AsQ3VycmVuY3k6OTgwLFB1cmNoYXNlVGltZT0iMTgwOTE3NDEyNiIsUHVyY2hhc2VEZXNjOiJUZlZXRva2Vu
In0=",
signature:"EXDEhK9kMK0lwTEWH4mm1oJvKm5vVFyXnyDnqEDHDc3mYyXEhLv3Ih6_fdmN-
apUPxgV5GEpV0YQWTuSyGF3o32dF0n-
A4LrZ93z8Dw7gj9ULLd5ffRE42x0tFL6jNNEnVUbj8WB1UeR6mRN4l4aTRaNU123hq6UIqB_jsTxWJU"}

```

Receive json response

```
{{{"header": "eyJhbGciOiJSUzUxMiJ9", "payload":
```

```
"eyJNZXJjaGFudEIEIjojMTc1Mjc3OSIsIiRlcm1pbmFsSUQiOiJFNzg4MDUzOSIsIkFwcHJvdmFsQ29kZSI6Ijc1
ODUwOCIsIlJybiI6IjgyNzUxMjczNjI4ODY4M0FDlixUb3RhbEFTb3VudDo1MDAsQ3VycmVuY3k6OTgwLFB1cmNoYXNlVGltZT0iMTgwOTE3NDEyNiIsUHVyY2hhc2VEZXNjOiJUZlZXRva2VuIn0=", "signature":
```

```
"iJucvSFqxVx6mCSSNfd3BqHBgjWEuWxxAbtdUKebj4LzNeJl_tAQG7Yqu-
```

```
tpL8c_Sm7DKkYu1Ehmi0NOgn4VP8_KM34d5E7wgpWYhIjEBl_By4Bcyex2MuMRzxukNDnWqYpFZXljbOb
m9gezS70rSoCcx6LHvdInW5LfdZY31Vo"}}
```

, that is decoded to

```
{"MerchantID": "1752739", "TerminalID": "E7880539", "ApprovalCode": "758508", "Rrn": "827512375132", "HostC
ode": "000", "TranCode": "000", "Comment": "Approved"}
```


Table 8

TRAN_CODE_ID	DESCRIPTION
0	Approved
101	Invalid card parameters
105	Not approved by emitent
108	Lost/stolen card
111	Non existent card
116	Insufficient funds
130	Limit is exceeded
290	Issuer is not accessible
291	Technical/Communication problem
401	Invalid format
402	Invalid Acquirer/Merchant data
403	Component communication failure
404	Authentication error
405	Signature is invalid
406	Quota of transactions exceeded
407	Merchant is not active
408	Transaction was not found
409	Too many transactions were found
410	The order was paid (possible replay)
411	The order request time is out-of-date
412	Replay order condition
413	Unknown card type
414	CVC required
420	The total amount of successful transactions per day is limited
421	Tran amount limit (non 3-D Secure full authenticated)
430	Transaction is prohibited by Gateway
431	Attempted 3D-Secure is not accepted
432	Card is in stop list
433	The number of transactions has exceeded the limit
434	The merchant does not accept cards from the country
435	CLient IP address is on stop list
436	The sum of amount transactions has exceeded the limit
437	The limit of card number inputs has been exceeded
438	Unacceptable currency code
439	The time limit from request to authorization has been exceeded
440	The authorization time limit has been exceeded
441	MPI interaction problem
442	ACS communication problem
450	Recurrent payments are prohibited
451	MPI service not enabled
452	Card-to-Card Payment service not enabled
460	Token service not enabled

501	Canceled by user
502	The web session is expired
503	Transaction was canceled by merchant
504	Transaction was canceled by gateway with reversal
505	Invalid sequense of operations
506	Preauthorized transaction is expired
507	Preauthorized transaction already processed with payment
508	Invalid amount to pay a preauthorized transaction
509	Not able to trace back to original transaction
510	Refund is expired
511	Transaction was canceled by settlement action
512	Repeated reversal or refund
601	Not completed
602	Waiting confirmation of instalment
902	Cannot process transaction
909	Cannot process transaction
999	transaction in progress..