4pay.net

4PAY.net Merchant’s integration procedure

API GATEWAY

Verified by VISA / Master Card Secure Code

FULL API GATEWAY

With this integration modality the Merchant manage, using his own technology, the communication between the customer and the 4Pay.net Credit Card Processor. The Merchant will collects Credit Card details and manage the payment result, using server to server messages to the 4Pay.net platform.

Let's see in details all payment phases:

- **Payment request**

When the customer, after the choice of the product / service from the Merchant's catalogue, decides to proceed with the payment and gives all credit card details, the Merchant's system should send an XML server to server message to require a new transaction to the 4Pay.net system, forwarding all details about the Credit Card. This message will contain the following parameters:

- Merchant's account identification value.
- Order identification (TRANSACTION_ID).
- Credit card details (number, exp. date and security code).
- Amount and currency of the transaction.
- Type of transaction. This one can have the following values:
  - "**Implicit accounting**" (AUTCONT); If the authorization is positive the 4Pay.net system will charge the amount to the customer's credit card and will accredit the same amount to the Merchant.
  - "**Explicit accounting**" (AUT); If the authorization is positive, the Merchant, **in a second time with an explicit communication to the 4Pay.net system**, requires to account the transaction charging the customer's credit card and accrediting the amount to himself.
- **TOKEN** (messages signature). During the communication process between the Merchant and the 4Pay.net Gateway system there is the possibility that a third person intercepts the message and tries to change the information. This risk is avoided introducing an authentication process through a "signature" on the messages transmitted.

The 4Pay.net payment gateway will control if all transaction's details are correct;

If all controls will be ok it will contact the credit card's Issuer to know if he adheres to the security protocols (Verified By Visa / MasterCard Secure Code).

- **Credit card holder authentication**

If the Credit Card issuer doesn't enroll to the Verified by Visa / MasterCard Secure Code service the payment can be made immediately; in this case 4Pay.net will send to the authorization systems the payment request and will return to the Merchant the transaction's result.

If the Credit Card issuer and the Customer adheres to the Verified by Visa / MasterCard Secure Code service, he must be redirected to the Issuer's application to be qualified. The redirection is made using an html page. The 4Pay.net platform will forward an HTML tag that the Merchant **should decode with BASE64 algorithm, and put the result to the customer's browser.** From this moment all the authentication process is managed by the Issuer which, at the end of the process, will redirect the customer's browser to an URL, arranged by the 4Pay.net platform, sending the authentication result.

The 4Pay gateway will elaborate the message received from the Issuer to verify if it is possible to continue with the payment. It is possible to continue with the payment only if the customer has entered the correct password.

- **Payment phase and result notification**

If the customer authentication is correct or the authentication was not required, the 4Pay.net gateway will send a payment request to the authorization systems and it will receive the related answer. In case of Verified by Visa / MasterCard Secure Code payment, the transaction result will be posted to the Merchant "**Response URL**" value specified in the 4Pay.net profile page. Otherwise the 4Pay.net gateway will send to the Merchant the result of the transaction, through the same connection used to received it. In both cases the response will contains the payment details if successful, or the reason why it has been refused (technical problems, authentication unsuccessful, authorization not possible, etc.).

- **Messages structure**

Integration with server to server call.
This are xml messages sent with POST method, in SSL modality, codified with the standard ISO-8859-15
In the following paragraphs we will describe detailed messages' fields.

- **Messages signature [TOKEN]**

Each message exchange between the 4Pay.net gateway and the Merchant's system will contain a **TOKEN** field. The TOKEN is a security string which gives the possibility to verify if the message comes to the awaited sender or if it was modified from a third person.

To generate a TOKEN will be use the following method:
the string resulted from the concatenation of all message's sensible parameters and one shared key [**Security Key**] from the 4Pay.net gateway and the Merchant.
The Merchant can find his personal Security Key in his 4Pay.net Profile page.

The token is calculated by concatenating the following information and performing an SHA-1 (Secure Hash Algorithm) hash on the resulting string:

- transaction_id [TEST0000000000000001]
- action_code [AUT]
- pan [4500000000000001]
- expire_date [any future date]
- amount [10]
- currency [EUR]
- your own security key i.e.: [SNJSNXCG7MF9BMZXPV63]

Concatenated String =
TEST0000000000000001AUT4500000000000001070710EURSNJSNXCG7MF9BMZXPV63
Token = 25a2d7e2d65a9a87efffd7efb6b787c3d84e269c
Please note that Authentication Token is NOT case-sensitive
If your account is temporary disabled, your security key may change after reactivation, please check it.

- **Integration by messages server to server**

This paragraph contains the messages details exchange between the 4Pay.net gateway and the Merchant's application.

New payment request.

Type of message : xml sent by http post in modality SSL

This is a request of a new payment that the Merchant’s application has to send to the 4Pay.net gateway by a call server to server with the following fields:

```
<?xml version="1.0" encoding="UTF-8"?>
<apcc>
<request>
<transaction_id>TEST0000000000000001</transaction_id>
<action_code>AUT</action_code>
<pan>4500000000000001</pan>
<expire_date>any future date</expire_date>
<cvv2>777</cvv2>
<amount>10</amount>
<currency>EUR</currency>
<description>TEST0000000000000001 order description</description>
<customer_ip>127.0.0.1</customer_ip>
<billing_email>customer@email.com</billing_email>
<billing_address>nowhere street, 234</billing_address>
<billing_city>New York</billing_city>
<billing_region>New Jersey</billing_region>
<billing_postal>10002</billing_postal>
<billing_country>US</billing_country>
<billing_phone>8005128547</billing_phone>
<billing_fname>John</billing_fname>
<billing_lname>Doe</billing_lname>
<bin_name>binname</bin_name>
<bin_phone>binphone</bin_phone>
</request>
<account>your@account.net</account>
<token>54077F031A9C8C967BB68B1BEED71A0CC1F801E1</token>
</apcc>
```

Message description:

PARAMETER NAME	M	Format	Example	Description
transaction_id	Y	alphanumeric, 20 characters fix	TEST0000000000000001	Merchant's order code
action_code	Y	Alphanumeric max 8 characters	AUT	Kind of required transaction. The following values are allowed: **AUT:** require a transaction with explicit accounting **AUT-CONT:** require a transaction with implicit accounting
pan	Y	alphanumeric, max 19 characters	1234567890123456	Credit card's number
expire_date	Y	Alphanumeric 4 characters fix Format : MMYY	0520 (to enter as expire date May 2020)	Credit Card expire date
cvv2	Y	Alphanumeric max 4 characters	1234	Credit card's security code
amount	Y	amount in Euro cents	125456 for 1254,56 usd	Transaction amount in Usd cents
currency	Y	Alphanumeric 3 characters fix	USD	Currency value code, see Appendix A1 for the possible values
description	Y	Alphanumeric, max 200 characters	Digital camera	Description order
customer_ip	Y	alphanumeric, 15 characters fix	208.69.34.231	Customer ip address
billing_email	Y	alphanumeric, 64 characters fix	customer@email.com	Customer email address
billing_address	Y	alphanumeric, 64 characters fix	Main street 1	Customer billing address
billing_city	Y	alphanumeric, 64 characters fix	Los Angeles	Customer billing city
billing_region	N	alphanumeric, 32 characters fix	California	Customer billing region/state
billing_postal	Y	alphanumeric, 32 characters fix	12345	Customer postal / ZIP code
billing_country	Y	characters, 2 characters fix	Unites States	Customer country
billing_phone	Y	alphanumeric, 32 characters fix	1234567456	Customer billing phone
billing_fname	Y	alphanumeric, 32 characters fix	John	Credit card holder first name
billing_lname	Y	alphanumeric, 32 characters fix	Doe	Credit card holder last name
bin_name	N	alphanumeric, 128 characters fix	Bank of America	Bank name that issued the credit card
bin_phone	N	alphanumeric, 128 characters fix	1234567456	Bank phone that issued the credit card
account	Y	alphanumeric, 128 characters fix	your@account.com	Merchant account
token	Y	alphanumeric, 32 characters fix	54077F031A9C8C967BB68B1BEED71A0CC1F801E1	Merchant token

M = mandatory / Y = yes / N = no

Response to new payment request

This message is returned from 4Pay.net gateway as an answer to the new payment request.

```
<?xml version="1.0" encoding="UTF-8"?>
<apcc>
<response>
<response_code>0</response_code>
<description>
Approved: Payment successful (TRANSACTION_ID: TEST0000000000000001) (AUTH_CODE:
006637) (TRANSACTION_TYPE: VBV_MERCHANT)
</description>
<authcode>006637</authcode>
<transaction_type>VBV_MERCHANT</transaction_type>
<transaction_id>TEST0000000000000001</transaction_id>
<transaction_reference>xxxxxxxxxxxxtest001</transaction_reference>
</response>
<token>4C844EC2BE168CF7C07A65F074F14543C82AD272</token>
</apcc>
```

Message description:

PARAMETER NAME	M	Format	Example	Description
response_code	Y	Alphanumeric max 3 characters	0	Result of therequired payment . For the possible values see the table below
description	Y	alphanumeric, 128 characters fix	Approved: Payment successful (TRANSACTION_ID: TEST0000000000000001) (AUTH_CODE: 006637) (TRANSACTION_TYPE: VBV_MERCHANT)	Payment long description
authcode	Y	Alphanumeric max 10 characters	006637	Authorization code sent to the Merchant from the credit card's issuer. In case of declined payment it will be sent as an empty string.
transaction_type	Y	Alphanumeric max. 20 characters	VBV_MERCHANT	Type of transaction. It shows the payment security level. See appendix A2 for the possible values
transaction_id	Y	Alphanumeric 20 characters fix	TEST0000000000000001	Same value as in the payment request message
transaction_reference	Y	Alphanumeric max. 20 characters	xxxxxxxxxxxxtest001	Unique Merchant transaction reference
token	Y	alphanumeric, 32 characters fix	54077F031A9C8C967BB68B1BEED71A0CC1F801E1	4Pay.net Response Token

M = mandatory
Y = yes
N = no

The **4Pay.net Response Token** is calculated by concatenating the following information and performing an SHA-1 hash on the resulting string:

- response_code
- description
- your own security key

Response URL parameter:

Our payment gateway supports "Verified by VISA" and "MasterCard Secure Code" features. In case you receive <html> tag as response, you should decode the tag data with BASE64 algorithm, and put the result to the customer's browser. After VbV/MSC check, the resulted XML will be send/POSTed in your "Response URL" address.

RESPONSE: Result of the required payment.
Response_code can have the following values:

Code	Description
0	Payment successful
1	Payment error: wrong message's format or missing field
2	Payment error: Unexpected error during the request or declined due to high risk factor
3	Payment error: TRANSACTION_ID field duplicated or not found
8	Payment error: wrong TOKEN
16	Payment error: Unexpected error during the request
17	Max number of transaction possible
18	Payment error: Payment refused from the credit card's issuer
19	Payment error: Authentication VbV/SC unsuccessful, payment not sent
20	Payment error: Authentication VbV/SC not possible, payment not sent
21	Transaction Error: Field TRANSACTION_ID unknown
22	Error in the transaction: transaction is not possible

Dispositive operations

Dispositive operations are messages to require the accounting or cancellation of a transaction which payment is positive.

The message must have the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
<apcc>
<request>
<transaction_id>TEST0000000000000001</transaction_id>
<action_code>CHARGE</action_code>
<auth_code>006637</auth_code>
<amount>100</amount>
<amount_op>100</amount_op>
<currency>USD</currency>
</request>
<account>your@account.net</account>
<token>ef24ae8902a14801950013d846028bcc70a1befe</token>
</apcc>
```

Message description:

PARAMETER NAME	M	Format	Example	Description
transaction_id	Y	Alphanumeric 20 characters fix	TEST0000000000000001	Univocal code of the order given by the Merchant
action_code	Y	Alphanumeric max 6 character	CHARGE	Type of transaction required. For the possible values look at the table below
auth_code	Y	Alphanumeric max 10 characters	006637	Authorization code received from 4pay gateway as answer to the payment request
amount	Y	amount in USD cents	125456 for 1254,56 usd	Amount for which the payment authorization was required
amount_op	Y	amount in USD cents	125456 for 1254,56 usd	Amount to charge/void/refund
currency	Y	Alphanumeric 3 characters fix	USD	Payment value code. Same value as in the payment request message
account	Y	alphanumeric, 128 characters fix	your@account.com	Merchant account
token	Y	alphanumeric, 32 characters fix	54077F031A9C8C967BB68B1BEED71A0CC1F801E1	Merchant token

M = mandatory
Y = yes
N = no

TOKEN: The following parameters are used for the TOKEN calculation:

- transaction_id
- action_code
- auth_code
- amount
- amount_op
- currency
- your own security key i.e.: [SNJSNXCG7MF9BMZXPV63]

Action_code:
The dispositive operations managed from the 4Pay.net gateway using this message are the following:

Code	Description
VOID	**Cancel the transaction** With this transaction we will put in reinstatement the plafond of the used credit card if the credit card's issuer will support this functionality. The amount to cancel must be the same or less as the amount request for the payment. *[ONLY AFTER AN **AUT** ACTION_CODE]*
CHARGE	**Accounting** This transaction will confirm the before authorized transaction charging the credit card of the buyer and crediting the amount to the Merchant. The amount to account must be the same or less as the amount request for the payment. *[ONLY AFTER AN **AUT** ACTION_CODE]*
REFUND	**Cancel the accounting** This transaction will charge the Merchant and will reaccredit the amount on the credit card of the buyer. The amount to cancel must be the same or less as the amount request for the payment. *[ONLY AFTER AN **AUT-CONT** ACTION_CODE OR **AUT -> CHARGE** CODES]*

Response to new dispositive operation request
This message is returned from 4Pay.net gateway as an answer to the new dispositive operation request.

```
<apcc>
<response>
<response_code>0</response_code>
<description>
Payment successful
</description>
</response>
<token>CC798F403FF2311E2C3ECFD2C21DACB7AE35DE98</token>
</apcc>
```

RESPONSE: Result of the required payment.
Response_code can have the following values:

Code	Description
0	Payment successful
1	Payment error: wrong message's format or missing field
2	Payment error: Unexpected error during the request
3	Payment error: TRANSACTION_ID field duplicated or not found
8	Payment error: wrong TOKEN
16	Payment error: Unexpected error during the request
18	Payment error: Payment refused from the credit card's issuer
21	Transaction Error: Field TRANSACTION_ID unknown
22	Error in the transaction: transaction is not possible

Appendix A1

The following table shows the codes available in the CURRENCY filed:

Code	Description
USD	US dollar
EUR (1)	Euro
JPY (1)	Japanese yen
BGN (1)	Bulgarian lev
CYP (1)	Cyprus pound
CZK (1)	Czech koruna
DKK (1)	Danish krone
EEK (1)	Estonian kroon
GBP (1)	Pound sterling
HUF (1)	Hungarian forint
LTL (1)	Lithuanian litas
LVL (1)	Latvian lats
MTL (1)	Maltese lira
PLN (1)	Polish zloty
RON (1)	New Romanian leu
SEK (1)	Swedish krona
SKK (1)	Slovak koruna
CHF (1)	Swiss franc
ISK (1)	Icelandic krona
NOK (1)	Norwegian krone
HRK (1)	Croatian kuna
TRY (1)	New Turkish lira [3]
AUD (1)	Australian dollar
CAD (1)	Canadian dollar
CNY (1)	Chinese yuan renminbi
HKD (1)	Hong Kong dollar
IDR (1)	Indonesian rupiah
KRW (1)	South Korean won
MYR (1)	Malaysian ringgit
NZD (1)	New Zealand dollar
PHP (1)	Philippine peso
SGD (1)	Singapore dollar

(1) Note: for using multicurrency system please contact 4Pay staff.

Appendix A2

The following table shows the code in the TRANSACTION_TYPE field and the transaction security level:

Code	Description
VBV_FULL	Credit card’s owner is registered for the service Verified by Visa and has putted the right code.
SC_FULL	Credit card’s owner is registered for the service Secure Code and has putted the right code.
VBV_MERCHANT	Credit card’s issuing company don’t take part to the service Verified by Visa.
SC_MERCHANT	Credit card’s issuing company don’t take part to the service Secure Code.
NO_3DSECURE	The proprietor is not qualified to use the security protocols Verified by Visa and Secure Code or it was not possible to use the protocols.

Test environment of Automation Interface

During integration test requests could be sent to

https://api.4pay.net/xmlapi.php

In test environment system does not create any transaction. Test requests must include the parameter <test>true</test> just after the <token> parameter.

Sample of TEST charge request:

```
<?xml version="1.0" encoding="UTF-8"?>
<apcc>
<request>
<transaction_id>TEST0000000000000001</transaction_id>
<action_code>AUT</action_code>
<pan>4539990000000038</pan>
<expire_date> any future date</expire_date>
<cvv2>123</cvv2>
<amount>10</amount>
<currency>USD</currency>
<description>TEST0000000000000001 order description</description>
</request>
<account>your@account.net</account>
<token>54077F031A9C8C967BB68B1BEED71A0CC1F801E1</token>
<test>true</test>
</apcc>
```

Test parameters:

Credit Card Number: 4539990000000038
Expiration date: any, later than the present day
Cvv2: any

Test Features:

Amount from $0 to $4.99: approved
Amount from $5 to $9.99: Verified by VISA check
Amount from $10: declined