



# Paynet - Internet Payment System

*Integration guidelines*

2.31-6

---

# **Paynet - Internet Payment System**

Copyright © 2009 PaynetEasy

Preface .....	v
<b>1. Making Non-3D Sale (SMS) Transactions .....</b>	<b>1</b>
1.1. Pre-requisites .....	1
1.2. Initiating a transaction .....	1
1.3. Request authorization through control parameter .....	3
1.4. Interpreting Non-3D Response .....	4
1.5. Non-3D Sale (SMS) Test Mode .....	4
<b>2. Making 3D Sale (SMS) Transaction .....</b>	<b>6</b>
2.1. Pre-requisites .....	6
2.2. 3D Transaction Process Flow .....	6
2.3. Initiating a transaction .....	6
2.4. Interpreting 3D Response .....	9
2.5. Interpreting redirect results .....	9
2.6. Interpreting server callback result .....	9
2.7. 3D Sale (SMS) Test Mode .....	10
<b>3. Making Pre-auth/Capture (DMS) Transaction .....</b>	<b>12</b>
3.1. Pre-requisites .....	12
3.2. Pre-auth Transaction Process Flow .....	12
3.3. Making Non-3D Pre-auth Transaction .....	13
3.4. Making 3D Pre-auth Transaction .....	13
3.5. Making Capture Transaction .....	13
3.5.1. Initiating a capture transaction .....	13
3.5.2. Interpreting Capture Response .....	13
<b>4. Making Refund Transaction .....</b>	<b>15</b>
4.1. Pre-requisites .....	15
4.2. Initiating a transaction .....	15
4.3. Interpreting Refund Response .....	16
<b>5. Advanced services .....</b>	<b>18</b>
5.1. Pre-requisites .....	18
5.2. Merchant URL callbacks .....	18
5.2.1. Sale and Refund callback simple URL .....	18
5.2.2. Sale and Refund callback Customized URL .....	19
5.2.2.1. Callback authorization through control parameter .....	20
5.2.3. Chargeback callback URL .....	20
5.3. Test mode .....	21
5.4. Order status API call .....	21
5.4.1. Order status API call results .....	22
5.4.2. Order status API call authorization through control parameter .....	23
5.5. Rebill API call .....	23
5.5.1. Rebill API call response .....	24
5.5.2. Rebill API call authorization through control parameter....	24

---

5.6. Transaction Report API .....	25
5.6.1. Transaction Report API call authorization through hash parameter .....	26
A. Two-Letter Country Codes .....	28
B. State Codes .....	36
C. Supported Currencies .....	39

---

# Preface

These Guidelines are aimed at specifying interface for integration into Paynet System. Paynet System is software intended for real-time payment processing over the Internet. It has a plain and user-friendly HTTP POST based interface for Merchants that can be easily implemented on any platform (PHP/Perl/Ruby, etc.).

---

# Chapter 1. Making Non-3D Sale (SMS) Transactions

## 1.1. Pre-requisites

1. Merchant must set up a merchant account in Paynet System and register his site. Any site in the System is identified by sid. (See Table 1.1, “Non-3D Payment Request Parameters”).
2. Merchant’s site is connected to the processing gate, what is actually done by Paynet System manager
3. In addition, Merchant may utilize a merchant control key for payment request authorization.
4. Paynet System manager specifies HTTP URL where Non-3D payment requests are to be sent. A typical URL has a format similar to the following: <https://specifieddomain.com/paynet/payment.html>

## 1.2. Initiating a transaction

In order to initiate a non-3D sale transaction, it is necessary to make an HTTP POST request with the parameters specified in Table 1.1, “Non-3D Payment Request Parameters”.

**Table 1.1. Non-3D Payment Request Parameters**

Parameter name	Length/ Type	Comment	Mandatory/ Optional
version	10/String	<i>Version number of the API being used. Version number must be equal to "1" for non-3D sale transactions.</i>	Mandatory
sid	12/ Numeric	<i>Site identifier (example: 3456)</i>	Mandatory
client_orderid	12/ Numeric	<i>Merchant order identifier.</i>	Mandatory
order_desc	125/ String	<i>Brief order description</i>	Mandatory
first_name	50/String	<i>Customer’s first name</i>	Mandatory
last_name	50/String	<i>Customer’s last name</i>	Mandatory

<b>Parameter name</b>	<b>Length/ Type</b>	<b>Comment</b>	<b>Mandatory/ Optional</b>
birthday	6/ Numeric	<i>Customer's date of birth, in the format MMDDYY.</i>	Optional
address1	50/String	<i>Customer's address line 1.</i>	Mandatory
city	50/String	<i>Customer's city.</i>	Mandatory
state	2/String	<i>Customer's state (two-letter US state code). Please see Appendix A for a list of valid US state codes. Not applicable outside the US.</i>	Optional
zip_code	10/String	<i>Customer's ZIP code.</i>	Mandatory
country	2/String	<i>Customer's country (two-letter country code). Please see Appendix B for a list of valid country codes.</i>	Mandatory
phone	15/String	<i>Customer's full international phone number, including country code.</i>	Mandatory
cell_phone	15/String	<i>The customer's full international cell phone number, including country code.</i>	Optional
email	50/String	<i>Customer's email address.</i>	Mandatory
amount	10/ Numeric	<i>Amount to be charged. The amount has to be specified in the highest units with "." delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents</i>	Mandatory
currency	3/String	<i>The currency the transaction is charged in (three-letter currency code). Valid parameter values are:</i> <ul style="list-style-type: none"> <li>• USD for US Dollar</li> <li>• EUR for European Euro</li> </ul>	Mandatory
credit_card_number	20/ Numeric	<i>Customer's credit card number.</i>	Mandatory
expire_month	2/ Numeric	<i>Credit card expiration month</i>	Mandatory
expire_year	2/ Numeric	<i>Credit card expiration year</i>	Mandatory
cvv2	3-4/ Numeric	<i>Customer's CVV2 code. CVV2 (Card Verification Value) is a three- or four-digit</i>	Mandatory

Parameter name	Length/ Type	Comment	Mandatory/ Optional
		<i>number AFTER the credit card number in the signature area of the card.</i>	
ipaddress	20/String	<i>Customer's IP address, included for fraud screening purposes.</i>	Mandatory
site_url	128/ String	<i>URL the original sale is made from.</i>	Optional
control	40/String	<i>Checksum generated by SHA-1.</i>	Optional

Please note the following characters must be escaped in the parameter values: '&', '+', '\.'

### 1.3. Request authorization through `control` parameter

The checksum is used to ensure that it is a particular Merchant (and not a fraudster) that initiates the transaction. This SHA-1 checksum, the parameter `control`, is created by concatenation of the parameters values in the following order:

- sid
- client\_orderid
- credit\_card\_number
- amount (in cents)
- email
- merchant\_control

A complete string example may look as follows:

```
59I09876544445555666611112526email@client.com3E8E45B5-7682-42D8-6ECC-
FB794F6B11B1
```

Encrypt the string using SHA-1 algorithm. The resultant string yields the `control` parameter (see Table 1.1, "Non-3D Payment Request Parameters") which is required for request authorization. For the above-mentioned example the `control` will take the following value:

```
5b1da0a20a1b9f350f4d66caaba15a9533e7ee13
```



## 1.4. Interpreting Non-3D Response

Upon completion by the System of Non-3D request processing, it returns the result with the following parameters:

**Table 1.2. Non-3D Response parameters**

Parameter Name	Parameter Description
status	The status code of the order. It may take value either "approved" or "declined"
error_message	This parameter contains error description in case <code>status</code> parameter equals "declined"
error_code	Error code that identifies the error.
orderid	Unique number assigned to the order for identification purposes.
client_orderid	Unique number of the order assigned by merchant site descriptor
descriptor	Payment descriptor of the gate through which the transaction has been processed.

All parameters are concatenated in the following way:

```
parameterName1=parameterValue1&parameterName2=parameterValue2
```

A typical response is shown in the example below:

```
status=declined&error_message=Decline, refer to card
issuer&error_code=107&orderid=S279G323P4T1209294&client
_orderid=c258d6536ababe65
```

## 1.5. Non-3D Sale (SMS) Test Mode

Merchant may choose to run test transactions before going live.

For Non-3D transactions Merchant may use the following dummy date for credit card and :

**Table 1.3. Test transaction parameters**

Transaction Parameter	Value
Credit card number	4444 5555 6666 1111

---

<b>Transaction Parameter</b>	<b>Value</b>
Credit card expiry date	Any valid date in future
CVV2	123 for approve, any other valid CVV2 for decline

For more details about test mode see Section 5.3, "Test mode"

---

# Chapter 2. Making 3D Sale (SMS) Transaction

## 2.1. Pre-requisites

1. Merchant must set up a merchant account in Paynet System and register his site. Any site in the System is identified by `sid`. (See Table 2.1, “3D Payment Request Parameters”).
2. Merchant’s site is connected to the processing gate, what is actually done by Paynet System manager.
3. . In addition, Merchant may utilize a `merchant_control` for payment request authorization.
4. Paynet System manager specifies HTTP URL where 3D payment requests are to be sent. A typical URL has a format similar to the following: `https://specifieddomain.com/paynet/payment.html`

## 2.2. 3D Transaction Process Flow

1. Merchant Site initiates a transaction by sending HTTP Post request to the specified URL (for instance, `https://specifieddomain.com/paynet/payment.html`)
2. The System processes the transaction and sends an HTML response which must be passed through to the cardholder browser.
3. Cardholder is redirected to `redirect_success_url` after valid response code is obtained from bank. The code can denote either the decline or approval. The process is finished.
4. Cardholder is redirected to `redirect_failure_url` when something failed on a bank’s side. The process is finished.
5. System additionally sends transaction results to `server_callback_url` in order to cover the cases where cardholder does not finish 3D authorization process properly.

## 2.3. Initiating a transaction

In order to initiate a 3D sale transaction, it is necessary to make an HTTP POST request with the parameters specified in Table 2.1, “3D Payment Request Parameters”.

**Table 2.1. 3D Payment Request Parameters**

<b>Parameter name</b>	<b>Length/ Type</b>	<b>Comment</b>	<b>Mandatory/ Optional</b>
version	10/String	<i>Version number of the API being used. Version number must be equal to "3" for 3D sale transactions.</i>	Mandatory
sid	12/ Numeric	<i>Site identifier (example: 3456)</i>	Mandatory
client_orderid	12/ Numeric	<i>Merchant order identifier.</i>	Mandatory
order_desc	125/ String	<i>Brief order description</i>	Mandatory
first_name	50/String	<i>Customer's first name</i>	Mandatory
last_name	50/String	<i>Customer's last name</i>	Mandatory
ssn	4/ Numeric	<i>Last four digits of the customer's social security number.</i>	Optional
birthday	6/ Numeric	<i>Customer's date of birth, in the format MMDDYY.</i>	Optional
address1	50/String	<i>Customer's address line 1.</i>	Mandatory
city	50/String	<i>Customer's city.</i>	Mandatory
state	2/String	<i>Customer's state (two-letter US state code). Please see Appendix A for a list of valid US state codes. Not applicable outside the US.</i>	Optional
zip_code	10/String	<i>Customer's ZIP code</i>	Mandatory
country	2/String	<i>Customer's country(two-letter country code). Please see Appendix B for a list of valid country codes.</i>	Mandatory
phone	15/String	<i>Customer's full international phone number, including country code.</i>	Mandatory
cell_phone	15/String	<i>Customer's full international cell phone number, including country code.</i>	Optional
email	50/String	<i>Customer's email address.</i>	Mandatory
amount	10/ Numeric	<i>Amount to be charged. The amount has to be specified in the highest units with "." delimiter. For instance, 10.5 for USD means 10 US Dollars and 50 Cents</i>	Mandatory

Parameter name	Length/ Type	Comment	Mandatory/ Optional
currency	3/String	<i>Currency the transaction is charged in (three-letter currency code). Valid parameter values are:</i> <ul style="list-style-type: none"> <li>• USD for US Dollar</li> <li>• EUR for European Euro</li> </ul>	Mandatory
credit_card_number	20/ Numeric	<i>Customer's credit card number.</i>	Mandatory
expire_month	2/ Numeric	<i>Credit card expiration month</i>	Mandatory
expire_year	2/ Numeric	<i>Credit card expiration year</i>	Mandatory
cvv2	3-4/ Numeric	<i>Customer's CVV2 code. CVV2 (Card Verification Value) is a three- or four-digit number AFTER the credit card number in the signature area of the card.</i>	Mandatory
ipaddress	20/String	<i>Customer's IP address, included for fraud screening purposes.</i>	Mandatory
site_url	128/ String	<i>URL the original sale is made from.</i>	Optional
control	40/String	<i>Checksum generated by SHA-1. See Section 1.3, "Request authorization through control parameter" for more details.</i>	Optional
redirect_success_url	128/ String	<i>URL the cardholder will be redirected to upon successful completion of the transaction. Please note that the cardholder will be redirected in any case, no matter whether the transaction is approved or declined.</i>	Optional
redirect_failure_url	128/ String	<i>URL the cardholder will be redirected to upon any failure on the bank's side. This is not an ordinary situation and it fully depends on the bank's behavior when this redirection is performed</i>	Optional
server_callback_url	128/ String	<i>URL the transaction result will be sent to. This callback data must be used in case</i>	Optional

Parameter name	Length/ Type	Comment	Mandatory/ Optional
		<i>the cardholder stops authorization process being on the bank-issuer's site.</i>	

Please note the following characters must be escaped in the parameter values: '&', '+', '\\'.

## 2.4. Interpreting 3D Response

The System returns HTML page which must be passed through without any changes to the client's browser.

## 2.5. Interpreting redirect results

Upon completion by the cardholder 3D authorisation process it is automatically redirected to `redirect_success_url`. The redirection is performed as an HTTP POST request with the parameters specified in Table 2.2, "3D Callback Parameters"

When 3D authorization process fails for bank reason, cardholder is automatically redirected to `redirect_failure_url`. The redirection is performed as HTTP POST request with the parameters specified in Table 2.2, "3D Callback Parameters"

## 2.6. Interpreting server callback result

Upon completion by the System of 3D request processing it returns the result on the specified `server_callback_url` with the following parameters:

**Table 2.2. 3D Callback Parameters**

Parameter Name	Parameter Description
status	The status code of the order. It may take value either "approved" or "declined" value
error_message	This parameter contains error description in case <code>status</code> parameter equals "declined"
error_code	Error code that identifies the error.
orderid	Unique number assigned to the order for identification purposes.
client_orderid	Unique number of the order assigned by merchant site.

Parameter Name	Parameter Description
descriptor	Payment descriptor of the gate through which the transaction has been processed.
control	The checksum generated by SHA-1. See below for more details.

The checksum is used to ensure that the callback is initiated for a particular Merchant, and not for anybody else claiming to be such Merchant. This SHA-1 checksum, the `control` parameter, is created by concatenation of the parameters values in the following order:

- status
- orderid
- client\_orderid
- merchant\_control

A complete string example may look as follows:

```
approvedS279G323P4T1209294c258d6536ababe653E8E45B5-7682-42D8-6ECC-
FB794F6B11B1
```

Encrypt the string using SHA-1 algorithm. The resultant string yields the `control` parameter. For the above-mentioned example the `control` will take the following value:

```
5b1da0a20a1b9f350f4d66caaba15a9533e7ee13
```

All parameters are sent via POST method. A typical response is shown in the example below::

```
status=declined&error_message=Decline, refer to card
issuer&error_code=107&orderid=S279G323P4T1209294&client
_orderid=c258d6536ababe65&descriptor=yoursale.com
```

## 2.7. 3D Sale (SMS) Test Mode

Merchant may choose to run test transactions before going "live".

For 3D Sale (SMS) transactions Merchant may use the following dummy credit card number:

**Table 2.3. Test transaction parameters**

<b>Transaction Parameter</b>	<b>Value</b>
Credit card number	4444 5555 6666 1111
Credit Card expiry date	Any valid date in future
CVV2	321 for "approve", any other valid CVV2 for "decline"

In the bank window you can enter "hint" in order to receive 3D Sale approval

For more details about test mode see Section 5.3, "Test mode"



---

# Chapter 3. Making Pre-auth/Capture (DMS) Transaction

## 3.1. Pre-requisites

1. Merchant must set up a merchant account in Paynet System and register his site. Any site in the System is identified by `sid`. See Table 2.1, “3D Payment Request Parameters”).
2. Merchant’s site is connected to the processing gate, what is actually done by Paynet System manager
3. In addition merchant MUST utilize a `merchant control` key for payment request authorization.
4. Paynet System manager specifies HTTP URL where 3D payment requests are to be sent. A typical URL has a format similar to the following: `https://specifieddomain.com/paynet/payment.html` - for pre-auth transaction  
`https://specifieddomain.com/paynet/capture.html` - for capture transaction

## 3.2. Pre-auth Transaction Process Flow

1. Merchant Site initiates a Non-3D/3D Pre-auth transaction by sending corresponding HTTP Post request to the specified URL (for instance, `https://specifieddomain.com/paynet/payment.html`)
2. The System processes the transaction and sends corresponding response (depending on 3D/Non-3D processing type). Upon successful completion of a pre-auth transaction the bank blocks the specified amount in the credit card account and does not allow the cardholder to use this blocked money. It is important to know that the block remains for a definite period of time depending on whether this is a debit or a credit card (usually the maximum block period is 7 days for debit cards and 28 days for credit cards).
3. Merchant Site sends Capture transaction in order to deduct the locked amount from credit card.
4. System processes capture transaction and return corresponding response. In this case the money is actually transferred from the bank-issuer account to the bank-acquirer account, which means the end of the transaction.

### 3.3. Making Non-3D Pre-auth Transaction

In order to make a non-3D pre-auth transaction it is necessary to perform the same steps described in Chapter 1, *Making Non-3D Sale (SMS) Transactions*. The only difference is to specify in `version` which must equal to "4".

### 3.4. Making 3D Pre-auth Transaction

In order to make a 3D pre-auth transaction it is necessary to perform the same steps described in Chapter 2, *Making 3D Sale (SMS) Transaction*. The only difference is to specify `version` which must equal to "6".

### 3.5. Making Capture Transaction

#### 3.5.1. Initiating a capture transaction

In order to initiate a capture transaction it is necessary to make HTTP POST request with the parameters specified in Table 3.1, "Capture Request Parameters"

**Table 3.1. Capture Request Parameters**

Parameter name	Length/Type	Comment	Mandatory/Optional
version	10/String	Version number of the API being used. Version number must be equal to "5" for capture transactions.	Mandatory
sid	12/ Numeric	Site identifier (example: 3456)	Mandatory
orderid	32/String	Unique number of pre-auth transaction returned by paynet system. It is returned in the <code>orderid</code> parameter of 3D/Non-3D response	Mandatory

#### 3.5.2. Interpreting Capture Response

When system processes the request it will return the result with the following parameters:

**Table 3.2. Capture Response parameters**

Parameter Name	Parameter Description
status	The status code of the order. It may take value either "approved" or "declined" value
error_message	This parameter contains error description in case <code>status</code> parameter equals "declined"
error_code	Error code that identifies the error.
orderid	Unique number which is assigned to the order for identification purposes.
client_orderid	Unique number of the order assigned by merchant site

All parameters are concatenated in the following way:

```
parameterName1=parameterValue1&parameterName2=parameterValue2
```

A typical response is shown in the example below:

```
status=declined&error_message=Decline, refer to card  
issuer&error_code=107&orderid=S279G323P4T1209294&client  
_orderid=c258d6536ababe65
```

---

# Chapter 4. Making Refund Transaction

## 4.1. Pre-requisites

1. Merchant must set up a merchant account in Paynet System and register his site. Any site in the System is identified by `sid`. (See Table 1.1, “Non-3D Payment Request Parameters”).
2. Merchant’s site is connected to the processing gate, what is actually done by Paynet System manager.
3. In addition, Merchant MUST utilize a merchant control key for payment request authorization.
4. Paynet System manager specifies HTTP URL where Non-3D payment requests are to be sent. A typical URL has a format similar to the following:  
`https://specifieddomain.com/paynet/refund.html`

## 4.2. Initiating a transaction

In order to initiate a refund transaction, it is necessary to make an HTTP POST request with the parameters specified in Table 1.1, “Non-3D Payment Request Parameters”.

**Table 4.1. Refund Request Parameters**

Parameter name	Length/ Type	Comment	Mandatory/ Optional
version	10/String	<i>Version number of the API being used. Version number must be equal to "1" for refund transactions.</i>	Mandatory
sid	12/ Numeric	<i>Merchnat Site identifier (example: 3456)</i>	Mandatory
orderid	20/String	<i>Unique identifier of Sale (SMS) or Pre-auth/Capture (DMS) transaction assigned by Paynet system and returned in the <code>orderid</code> response parameter. Please note that refund can be made On a pre-auth transaction if the relative capture transaction has not taken place yet; or</i>	Mandatory

Parameter name	Length/ Type	Comment	Mandatory/ Optional
		<i>on a capture transaction. In the latter case, processing of a refund on a pre-auth transaction is restricted.</i>	
client_orderid	128/ String	<i>Unique merchant identifier of Sale (SMS) or Pre-auth/Capture (DMS) transaction which is send to Paynet system in the client_orderid parameter.</i>	Optional
comment	50/String	<i>A brief description of reason</i>	Optional
blacklist	1/String	<i>Add transaction details info to blacklist. Blacklist parameter must be equal "D" for addition to blacklist.</i>	Optional

Please note the following characters must be escaped in the parameter values: '&', '+', '\'.

### 4.3. Interpreting Refund Response

When system processes the refund it will return the result with the following parameters:

**Table 4.2. Refund Response parameters**

Parameter Name	Parameter Description
status	The status code of the order. It may take either "approved", "declined" or "pending" value. In case "pending" status is obtained Merchant should be aware that there were problems during the refund processing and completion of such refund is postponed until Paynet support settles the problem.
error_message	This parameter contains error description in case <code>status</code> parameter equals "declined"
error_code	Error code that identifies the error.
orderid	Unique number assigned to the order for identification purposes.
client_orderid	The unique number of the order assigned by merchant site

All parameters are concatenated in the following way:

```
parameterName1=parameterValue1&parameterName2=parameterValue2
```

A typical response is shown in the example below:

```
status=declined&error_message=Decline, refer to card  
issuer&error_code=107&orderid=S279G323P4T1209294&client  
_orderid=c258d6536ababe65
```

---

# Chapter 5. Advanced services

## 5.1. Pre-requisites

1. Merchant must set up a merchant account in Paynet System and register his site. Any site in the System is identified by `sid`. (See Table 1.1, “Non-3D Payment Request Parameters”).
2. Site is connected to processing gate what is actually done by Paynet system manager
3. In addition merchant may require to have a merchant control key which is used for payment request authorization

## 5.2. Merchant URL callbacks

When Merchant creates site in Merchant console the following callback URLs might be defined:

- Sale callback URL
- Refund callback URL
- Chargeback callback URL

These callback URLs will be called when the transaction of respective type(i.e. sale,refund or chargeback) is completed, whether approved or declined. This gives a Merchant better control how the transaction is processed on Merchant's side, for example to add appropriate records to Merchant's internal accounting system.

### 5.2.1. Sale and Refund callback simple URL

Simple form of sale or refund callback is just a URL to Merchant's target page or script without any parameters, for example `http://www.i-cool-merchant.com/sale.php`. In this case the system automatically adds the following parameters to callback URL

**Table 5.1. Sale, Refund Callback Parameters**

Parameter name	Parameter Description
status	Transaction status, <i>approved</i>   <i>declined</i>   <i>processing</i>
orderid	Merchant order identifier, <i>client_orderid</i>
txn_id	Paynet transaction id
type	Transaction type, <i>sale</i>   <i>refund</i>

Parameter name	Parameter Description
amount	Transaction amount
name	Cardholder Name
email	Customer's email
comment	Comment in case of <i>refund</i> transaction
merchantdata	Reserved

## 5.2.2. Sale and Refund callback Customized URL

Customized callback URL is a fully defined URL with all the parameters Merchant's target page or script would require. Customized URL allows defining Merchant's own parameter names, whereas the actual parameters values are defined by use of macros with the following format `${parameter_name}`. Thus Paynet substitutes respective parameter values into Customized URL before calling it.

Example: `http://www.i-cool-merchant.com/sale_completed.php?cardholder_name=${name}&tx_status=${status}&order_id=${merchant_order}`

**Table 5.2. Sale, Refund Callback Macros**

Parameter name	Description
<code>\${status}</code>	Transaction status, <i>approved</i>   <i>declined</i>   <i>processing</i>
<code>\${merchant_order}</code>	Merchant order identifier, <i>client_orderid</i>
<code>\${orderid}</code>	Paynet transaction id
<code>\${type}</code>	Transaction type, <i>sale</i>   <i>refund</i>
<code>\${amount}</code>	Transaction amount
<code>\${descriptor}</code>	Payment descriptor of the gate through which the transaction has been processed.
<code>\${error_message}</code>	Error message: when <code>\${status}</code> meaning <i>declined</i>
<code>\${name}</code>	Cardholder Name
<code>\${email}</code>	Customer's email
<code>\${comment}</code>	Comment in case of <i>refund</i> transaction
<code>\${control}</code>	Checksum is used to ensure that it is Paynet (and not a fraudster) that initiates the callback for a particular Merchant. This is SHA-1 checksum of the concatenation <code>status+orderid+merchant_order+merchant_control</code> . The callback script MUST check this parameter by comparing it to SHA-1 checksum of the above concatenation. See Section 5.2.2.1, "Callback authorization through control



Parameter name	Description
	parameter” for more details about generating control checksum.
<code>#{merchantdata}</code>	Reserved

### 5.2.2.1. Callback authorization through `control` parameter

The checksum is used to ensure that the callback is sent to the merchant by Paynet, and not by a fraudster.

WARNING! If the Merchant does not check `control` parameter in the callback script a fraudster might use the callback URL to carry out fraudulent activities on the Merchant's system

This SHA-1 checksum, the parameter `control`, is created by concatenating of the values of the parameters in the following order:

- `status`
- `orderid`
- `merchant_order`
- `merchant_control`

The complete string example may look as follows:

```
59I098765444455556666111125263E8E45B5-7682-42D8-6ECC-FB794F6B11B1
```

Encrypt the string using SHA-1 algorithm. The resultant string yields the `control` parameter which is required for authorizing the callback. For the example the `control` above will take the following value:

```
5b1da0a20a1b9f350f4d66caaba15a9533e7ee13
```

### 5.2.3. Chargeback callback URL

Chargeback callback is a URL to Merchant's target page or script to handle chargeback transaction, for example to substitute chargeback amount in Merchant's accounting system. Chargeback callback is only available in simple form, for example `http://www.i-cool-merchant.com/chargeback.php`. The following parameters are automatically added to callback URL:

**Table 5.3. Chargeback Callback Parameters**

Parameter name	Description
<code>chargeback_order_id</code>	Chargeback transaction id

Parameter name	Description
invoiceno	Merchant order identifier, <i>client_orderid</i>
sale_order_id	Original Paynet transaction id the chargeback was requested for
amount	Transaction amount
comment	Reason for chargeback

### 5.3. Test mode

Upon integration of Merchant with Paynet it is strongly recommended to run a number of test transactions. Test sale transactions can be run with the use of dummy credit card numbers for both Non-3D and 3D Sale (SMS).

See Section 1.5, “Non-3D Sale (SMS) Test Mode” and Section 2.7, “3D Sale (SMS) Test Mode” for more details.

To run transactions in test mode Merchant must follow these steps:

1. It is up to Merchant to perform any testing but Merchant should inform Paynet Support if Merchant decides to do the testing before going live
2. Testing must be carried out from 10 a.m. to 6 p.m. GMT+3 (Moscow time) to ensure that technical support staff is available
3. Merchant must provide Paynet staff with an e-mail address to be used as Client Account in the Paynet ticket center
4. After testing is complete Merchant should contact Paynet staff no later than 5 p.m. GMT+3 (Moscow time) in order to initiate go live for the gate.

### 5.4. Order status API call

Merchant may use Order status API call to get the client's order transaction status.

To get such status Merchant should create the query string and call

`http://PAYNET-SERVER/paynet/status.html`, where `PAYNET-SERVER` is the server running Paynet, for example `http://paynet-cool-host.biz/paynet/status.html`

**Table 5.4. Order status API call parameters**

Parameter	Description
login	Merchant login name

Parameter	Description
client_orderid	Merchant order identifier of the transaction for which the status is requested
orderid	Order id assigned to the order by Paynet
control	Checksum used to ensure that it is Paynet (and not a fraudster) that initiates the callback for a particular Merchant. This is SHA-1 checksum of the concatenation <code>login+client_orderid+orderid+merchant_control</code> . See Section 5.4.2, "Order status API call authorization through control parameter" for more details about generating control checksum.

### 5.4.1. Order status API call results

The result is returned as a string similar to  
`status=declined&error_message=Card has expired&error_code=0005&orderid=12388`

**Table 5.5. Order status API call results parameters**

Parameter	Description
status	The status code of the order. May be approved OR declined
error_message	If status is declined this parameter contains the reason for decline
error_code	The error code is case of declined status
orderid	Order id assigned to the order by Paynet

For example Merchant makes a call

```
http://PAYNET-SERVER/paynet/status.html?
login=cool_merchant&client_orderid=123&orderid=987&control=86cb6b236faac1e58cf76995d6d
```

. Paynet may return the following string as a result

```
status=declined&error_message=Card has
expired&error_code=0005&orderid=987
```

or in case of approved

```
status=approved&error_message=&error_code=&orderid=987
```

## 5.4.2. Order status API call authorization through `control` parameter

The checksum is used to ensure that it is Merchant (and not a fraudster) that sends the callback to Paynet. This SHA-1 checksum, the parameter `control`, is created by concatenating of the values of the parameters in the following order:

- login
- client\_orderid
- orderid
- merchant\_control

The complete string example may look as follows:

```
cool_merchant56244443333222211111109625r45a019070772d1c4c2b503bbdc0fa22
```

Encrypt the string using SHA-1 algorithm. The resultant string yields the `control` parameter which is required for authorizing the callback. For the example `control` above will take the following value:

```
5b1da0a20a1b9f350f4d66caaba15a9533e7ee13
```

## 5.5. Rebill API call

Merchant may use Rebill API call to rebill any client for the given order id. To rebill the transaction Merchant should create the query string and call `http://PAYNET-SERVER/paynet/rebill.html`, where `PAYNET-SERVER` is the server running Paynet, for example `http://paynet-cool-host.biz/paynet/rebill.html`

**Table 5.6. Rebill API call parameters**

Parameter Name	Parameter Description
version	1 for sale and 4 for preauth
sid	Site identifier (example: 3456)
client_orderid	Client order id for the transaction which needs to be rebilled
amount	Amount for new transaction.
control	Checksum used to ensure that it is Paynet (and not a fraudster) that

Parameter Name	Parameter Description
	initiated the callback for a particular Merchant. This is SHA-1 checksum of the concatenation <code>sid+client_orderid+amount (in cents)+merchant_control</code> . See Section 5.5.2, “Rebill API call authorization through control parameter” for more details about generating control checksum.

### 5.5.1. Rebill API call response

Example URL to rebill a transaction

```
http://PAYNET-SERVER/paynet/status.html?
version=1&sid=123&client_orderid=123&amount=29.99&control=86cb6b236faac1e58cf76995d6da
```

. Rebill API call returns exactly the same result string as for Non-3D Sale (SMS) Transaction. See Section 1.4, “Interpreting Non-3D Response” for details

### 5.5.2. Rebill API call authorization through `control` parameter

The checksum is used to ensure that it is Merchant (and not a fraudster) that sends the callback to Paynet. This SHA-1 checksum, the parameter `control`, is created by concatenation of the parameters values in the following order:

- `sid`
- `client_orderid`
- `amount (in cents)`
- `merchant_control`

The complete string example may look as follows:

```
12347654367005b1da0a20a1b9f350f4d66caaba15a9533e7ee13
```

Encrypt the string using SHA-1 algorithm. The resultant string yields the `control` parameter which is required for authorizing the callback. For the example the `control` above will take the following value:

```
5b1da0a20a1b9f350f4d66caaba15a9533e7ee13
```

## 5.6. Transaction Report API

Merchant may use Transaction Report API call to get transaction report for specified period in Microsoft Excel XML format.

To get the Transaction Report Merchant should create the query string and call `http://PAYNET-SERVER/paynet/services/TransactionsReport`, where `PAYNET-SERVER` is the server running Paynet, for example `http://paynet-cool-host.biz/paynet/services/TransactionsReport`

**Table 5.7. Transaction Report API call parameters**

Parameter	Description
login	Merchant login name
fromDate	Start date and time from which transactions are requested.
toDate	End date and time to which transactions are requested
sites	Comma-delimited list of SIDs of sites for which you would like to get transactions. If empty, transactions for all sites which belong to the Merchant will be returned.
transactionTypes	Comma-delimited list of transaction types which should be used to filter transactions. If empty, all transaction types are considered. Valid transaction types are: sale, refund, chargeback, preauth, capture, fraud.
transactionStatuses	Comma-delimited list of transaction statuses which should be used to filter transactions. If empty, all transaction statuses are considered. Valid transaction statuses are: approved, declined, pending, processing.
hash	The checksum used to ensure that it is Paynet and not a fraudster that initiates the callback for a particular Merchant. This is SHA-1 checksum of the concatenation <code>fromDate+toDate+transactionTypes+transactionStatuses+sites+login+merchant_control</code> . See Section 5.6.1, "Transaction Report API call authorization

Parameter	Description
	through hash parameter” for more details about generating control checksum.

Dates are passed in the following format: 2009-01-02 03:04:55. This means January, 2nd 2009, time is 03:04:55 A.M. GMT+3 (Moscow time).

All sites with the given SIDs must belong to Merchant with the specified login. Otherwise such request will be declined.

Period (from fromDate to toDate) must not be more than 31 days.

Merchant is allowed to request report maximum 10 times per hour.

Example URL to get the transaction report:

```
http://PAYNET-SERVER/paynet/services/TransactionsReport?
fromDate=2009-01-02+03:04:55&toDate=2009-01-12+00:00:00&
transactionTypes=sale,refund,chargeback&transactionStatuses=approved,declined,pending
sites=11,22,333&login=cool_merchant&hash=5b1da0a20a1b9f350f4d66caaba15a9533e7ee13
```

### 5.6.1. Transaction Report API call authorization through hash parameter

The checksum is used to ensure that it is Merchant (and not a fraudster) that sends the callback to Paynet. This SHA-1 checksum, the parameter `hash`, is created by concatenating of the values of the parameters in the following order:

- fromDate
- toDate
- transactionTypes
- transactionStatuses
- sites
- login
- merchant\_control

The complete string example may look as follows:

```
2009-01-02 03:04:552009-01-12
00:00:00sale,refund,chargebackapproved,declined,pending11,22,333cool_merchant5b1da0a2
```

Encrypt the string using SHA-1 algorithm. The resultant string yields the *control* parameter which is required for authorizing the callback. For the example the *hash* above will take the following value:

```
5b1da0a20a1b9f350f4d66caaba15a9533e7ee13
```



---

# Appendix A. Two-Letter Country Codes

**Table A.1. Two-Letter Country Codes**

<b>Country</b>	<b>Code</b>
Afghanistan	AF
Albania	AL
Algeria	DZ
American Samoa	AS
Andorra	AD
Angola	AO
Anguilla	AI
Antarctica	AQ
Antigua And Barbuda	AG
Argentina	AR
Armenia	AM
Aruba	AW
Australia	AU
Austria	AT
Azerbaijan	AZ
Bahamas	BS
Bahrain	BH
Bangladesh	BD
Barbados	BB
Belarus	BY
Belgium	BE
Belize	BZ
Benin	BJ
Bermuda	BM
Bhutan	BT
Bolivia	BO
Bosnia And Herzegovina	BQ

<b>Country</b>	<b>Code</b>
Botswana	BW
Bouvet Island	BV
Brazil	BR
British Indian Ocean Territory	IO
Brunei Darussalam	BN
Bulgaria	BG
Burkina Faso	BF
Burundi	BI
Cambodia	KH
Cameroon	CM
Canada	CA
Cape Verde	CV
Cayman Islands	KY
Central African Republic	CF
Chad	TD
Chile	CL
China	CN
Christmas Island	CX
Cocos (Keeling) Islands	CC
Colombia	CO
Comoros	KM
Congo	CG
Congo, The Democratic Republic Of The	CD
Cook Islands	CK
Costa Rica	CR
Cote D'Ivoire	CI
Croatia	HR
Cuba	CU
Cyprus	CY
Czech Republic	CZ
Denmark	DK
Djibouti	DJ

<b>Country</b>	<b>Code</b>
Dominica	DM
Dominican Republic	DO
East Timor	TP
Ecuador	EC
Egypt	EG
El Salvador	SV
Equatorial Guinea	GQ
Eritrea	ER
Estonia	EE
Ethiopia	ET
Falkland Islands (Malvinas)	FK
Faroe Islands	FO
Fiji	FJ
Finland	FI
France	FR
French Guyana	GF
French Guyana	GP
French Polynesia	PF
French Southern Territories	TF
Gabon	GA
Gambia	GM
Georgia	GE
Germany	DE
Ghana	GH
Gibraltar	GI
Greece	GR
Greenland	GL
Grenada	GD
Guadeloupe	GP
Guam	GU
Guatemala	GT
Guinea	GN

<b>Country</b>	<b>Code</b>
Guinea-Bissau	GW
Guyana	GY
Haiti	HT
Heard Island And McDonald Islands	HM
Holy See (Vatican City State)	VA
Honduras	HN
Hong Kong	HK
Hungary	HU
Iceland	IS
India	IN
Indonesia	ID
Iran, Islamic Republic Of	IR
Iraq	IQ
Ireland	IE
Israel	IL
Italy	IT
Jamaica	JM
Japan	JP
Jordan	JO
Kazakhstan	KZ
Kenya	KE
Kiribati	KI
Korea, Democratic People's Republic Of	KP
Korea, Republic Of	KR
Kuwait	KW
Kyrgyzstan	KG
Lao People's Democratic Republic	LA
Latvia	LV
Lebanon	LB
Lesotho	LS
Liberia	LR
Libyan Arab Jamahiriya	LY

<b>Country</b>	<b>Code</b>
Liechtenstein	LI
Lithuania	LT
Luxembourg	LU
Macau	MO
Macedonia	MK
Madagascar	MG
Malawi	MW
Malaysia	MY
Maldives	MV
Mali	ML
Malta	MT
Marshall Islands	MH
Martinique	MQ
Mauritania	MR
Mauritius	MU
Mayotte	YT
Mexico	MX
Micronesia, Federated States Of	FM
Moldova, Republic Of	MD
Monaco	MV
Mongolia	MN
Montserrat	MS
Morocco	MA
Mozambique	MZ
Myanmar	MM
Namibia	NA
Nauru	NR
Nepal	NP
Netherlands	NL
Netherlands Antilles	AN
New Caledonia	NC
New Zealand	NZ

<b>Country</b>	<b>Code</b>
Nicaragua	NI
Niger	NE
Nigeria	NG
Niue	NU
Norfolk Island	NF
Northern Mariana Islands	MP
Norway	NO
Oman	OM
Pakistan	PK
Palau	PW
Panama	PA
Papua New Guinea	PG
Paraguay	PY
Peru	PE
Philippines	PH
Pitcairn	PN
Poland	PL
Portugal	PT
Puerto Rico	PR
Qatar	QA
Reunion	RE
Romania	RO
Russian Federation	RU
Rwanda	RW
Saint Helena	SH
Saint Kitts And Nevis	KN
Saint Lucia	LC
Saint Pierre And Miquelon	PM
Saint Vincent And The Grenadines	VC
Samoa	WS
San Marino	SM
Sao Tome And Principe	ST

<b>Country</b>	<b>Code</b>
Saudi Arabia	SA
Senegal	SN
Seychelles	SC
Sierra Leone	SL
Singapore	SG
Slovakia	SK
Slovenia	SI
Solomon Islands	SB
Somalia	SO
South Africa	ZA
Spain	ES
Sri Lanka	LK
Sudan	SD
Suriname	SR
Svalbard And Jan Mayen	SJ
Swaziland	SZ
Sweden	SE
Switzerland	CH
Syrian Arab Republic	SY
Taiwan, Province Of China	TW
Tajikistan	TJ
Tanzania, United Republic Of	TZ
Thailand	TH
Togo	TG
Tokelau	TK
Tonga	TO
Trinidad And Tobago	TT
Tunisia	TN
Turkey	TR
Turkmenistan	TM
Turks And Caicos Islands	TC
Tuvalu	TV

---

<b>Country</b>	<b>Code</b>
Uganda	UG
Ukraine	UA
United Arab Emirates	AE
United Kingdom	GB
United States	US
United States Minor Outlying Islands	UM
Uruguay	UY
Uzbekistan	UZ
Vanuatu	VU
Venezuela	VE
Vietnam	VN
Virgin Islands, British	VG
Virgin Islands, U.S.	VI
Wallis And Futuna	WF
Western Sahara	EH
Yemen	YE
Yugoslavia	YU
Zambia	ZM
Zimbabwe	ZW



# Appendix B. State Codes

**Table B.1. State Codes**

<b>Country</b>	<b>State Code</b>	<b>State Name</b>
AU	NSW	New South Wales
AU	WA	Western Australia
AU	VIC	Victoria
AU	TAS	Tasmania
AU	SA	South Australia
AU	QLD	Queensland
CA	NS	Nova Scotia
CA	NB	New Brunswick
CA	ON	Ontario
CA	QC	Quebec
CA	MB	Manitoba
CA	SK	Saskatchewan
CA	NF	Newfoundland
CA	AB	Alberta
CA	BC	British Columbia
US	VA	Virginia
US	OR	Oregon
US	YT	Yukon
US	OK	Oklahoma
US	OH	Ohio
US	NU	Nunavut
US	NT	Northwest Terr.
US	ND	North Dakota
US	NC	North Carolina
US	PA	Pennsylvania
US	PE	Prince Edward Isl.
US	VT	Vermont
US	UT	Utah
US	TX	Texas

<b>Country</b>	<b>State Code</b>	<b>State Name</b>
US	TN	Tennessee
US	WA	Washington
US	SC	South Carolina
US	WV	West Virginia
US	RI	Rhode Island
US	WI	Wisconsin
US	WY	Wyoming
US	SD	South Dakota
US	NY	New York
US	IN	Indiana
US	IL	Illinois
US	ID	Idaho
US	HI	Hawaii
US	GA	Georgia
US	FL	Florida
US	DE	Delaware
US	DC	D.C.
US	CT	Connecticut
US	CO	Colorado
US	CA	California
US	AR	Arkansas
US	AZ	Arizona
US	AK	Alaska
US	IA	Iowa
US	KS	Kansas
US	KY	Kentucky
US	NM	New Mexico
US	NJ	New Jersey
US	NH	New Hampshire
US	NV	Nevada
US	NE	Nebraska
US	MT	Montana

---

<b>Country</b>	<b>State Code</b>	<b>State Name</b>
US	MO	Missouri
US	MS	Mississippi
US	MN	Minnesota
US	MI	Michigan
US	MA	Massachusetts
US	MD	Maryland
US	ME	Maine
US	LA	Louisiana
US	AL	Alabama

---

# Appendix C. Supported Currencies

**Table C.1. Supported Currencies**

<b>Currency Description</b>	<b>Currency Code</b>
European Euro	EUR
US Dollar	USD