

PayU S.A.

Marcelinska 90, 60-324 Poznań

tel. (0-61) 630 60 05, fax (0-61) 860 27 01

Platnosci.pl

**Technical documentation for
partners**

© 2005-2010 PayU S.A.

08-06-2010
Version: 1.55

Table of Contents

1. General assumptions.....	3
2. Terms and fixed values used in the application.....	4
2.1. Error codes.....	5
2.2. Transaction statuses.....	6
2.2.1 Additional information about transaction statuses.....	7
2.3. Parameters of a new payment.....	8
2.4. Payment methods.....	9
2.4.1. Test - t.....	9
3. Integration with Platnosci.pl.....	10
3.1. Configuration data.....	10
3.2. Structure of the addresses: UrlPositive, UrlNegative.....	10
3.3. URL addresses of Platnosci.pl applications and available procedures.....	11
3.3.1. Encoding.....	11
3.3.2. Data format.....	11
3.4. MD5 signatures.....	11
3.5. Creating a new payment.....	12
3.6. Signing parameters transferred to a new payment.....	12
3.7. Exchange of information regarding transactions.....	13
3.7.1. Notifying the Shop about changing transaction status.....	13
3.7.2. Reading transaction status.....	14
3.7.3. Receiving the payment.....	17
3.7.4. Rejecting the payment.....	17
3.7.5. Operation performance status.....	18
3.8. Exchanging transaction information using WEBAPI/SOAP.....	18
4. Premium SMS.....	19
4.1. New transaction parameters for Premium SMS.....	19
4.2. List of acceptable values for Premium SMS transaction.....	19
4.3 Creating new SMS transaction.....	20
4.4 Exchanging information about Premium SMS transactions.....	20
5. Website tools.....	21
5.1 Dynamic list of payment methods - JavaScript.....	21
5.2 Dynamic list of payment methods - xml.....	22
5.3 Platnosci.pl basket.....	23
5.3.1 Adding items to basket.....	23
5.3.2 Displaying basket content.....	24
6. Additional services.....	25
6.1 MassPay.....	25
6.1.1 File format with data for MassPay.....	25
7. Document changes history.....	26

1. General assumptions

We would like to present you a new product implemented by PayU S.A., called Platnosci.pl as a response to the growing interest in professional tools designed for collective control of payments for the sale of services and goods on the internet.

Platnosci.pl is a website aimed at all e-commerce platforms that require professional payment solutions for their activity. Platnosci.pl means fast, secure and simple payment methods for goods and services offered on the Internet in order to make the clients feel fully comfortable and convenient. Years of experience, supported by professional technical and legal support, allow us to offer an unmatched system for handling internet payments.

2. Terms and fixed values used in the application

Platnosci.pl	- an application responsible for payments settlement
Company	- an entity that uses Platnosci.pl mechanisms to collect funds from a client
Shop	- an internet shop collecting the payments; one company can own a couple of shops
Pos	- a customer service point collecting payments; all service parameters are defined for a specific customer service point; one Shop may have several customer service points.
Client	- a person making payments
UrlPlatnosci.pl	- URL address where the Platnosci.pl application is installed, https://www.platnosci.pl/paygw/
UrlPositive	- URL address of the Shop application where the Client will be redirected after a transaction starts in an appropriate way
UrlNegative	- URL address of the Shop application where the Client will be redirected after a transaction starts in an inappropriate way
UrlOnline	- URL address of the Shop application where information about payment status changes - reports will be sent, using the POST method

2.1. Error codes

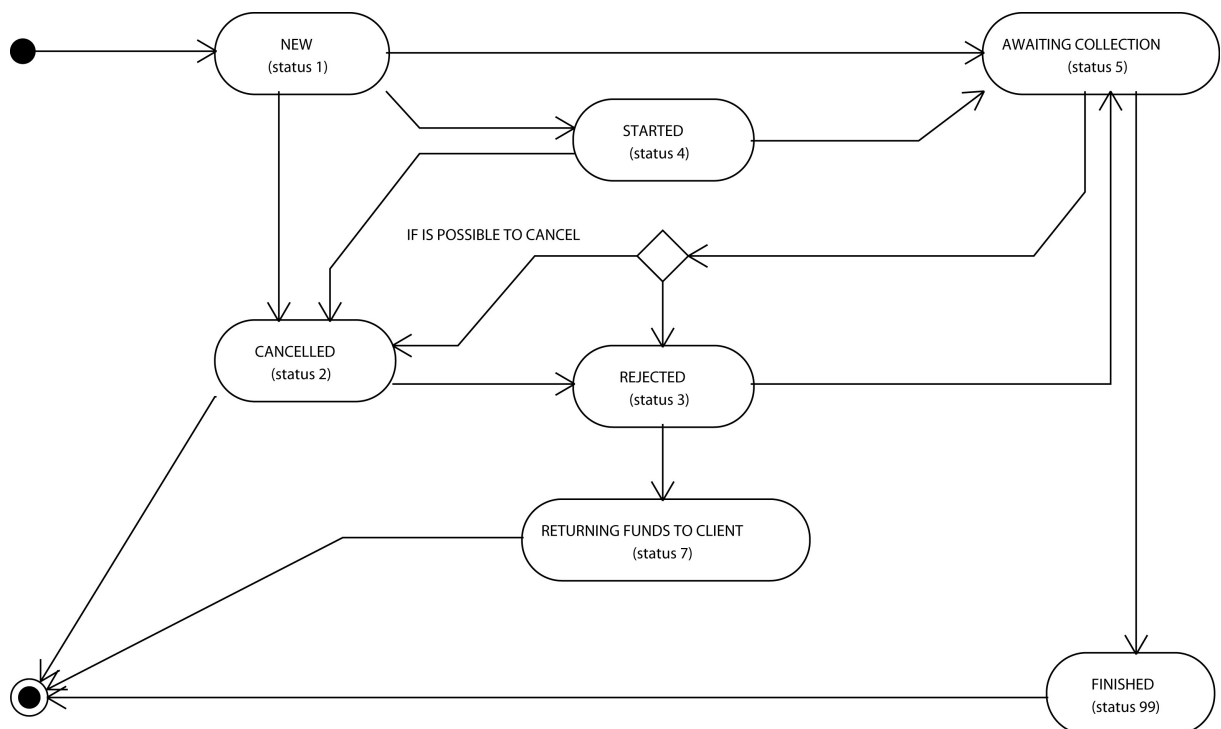
value	description
100	no or incorrect pos_id parameter
101	no session_id parameter
102	no ts parameter
103	no or incorrect sig parameter
104	no desc parameter
105	no client_ip parameter
106	no first_name parameter
107	no last_name parameter
108	no street parameter
109	no city parameter
110	no post_code parameter
111	no amount parameter
112	incorrect bank account number
113	no e-mail parameter
114	no phone number
200	other temporary error
201	other temporary database error
202	Pos of specified id is blocked
203	invalid pay_type value for given pos_id
204	specified payment method (pay_type value) is temporarily blocked for the given pos_id, e.g. maintenance break of payment gateway
205	transaction amount lower than the minimal value
206	transaction amount higher than the maximal value
207	exceeded value of all transactions for one client in the last period of time
208	Pos operates in ExpressPayment variant but this form of cooperation hasn't been activated (awaiting consent of customer support department)
209	incorrect pos_id or pos_auth_key number
500	no such transaction
501	no authorisation for specified transaction
502	transaction started earlier
503	transaction has already been authorised
504	transaction cancelled earlier
505	transaction previously forwarded for collection
506	transaction already collected
507	error while returning funds to client
599	incorrect transaction status, e.g. transaction cannot be acknowledged several times or other, please contact us
999	other critical error - please contact us

2.2. Transaction statuses

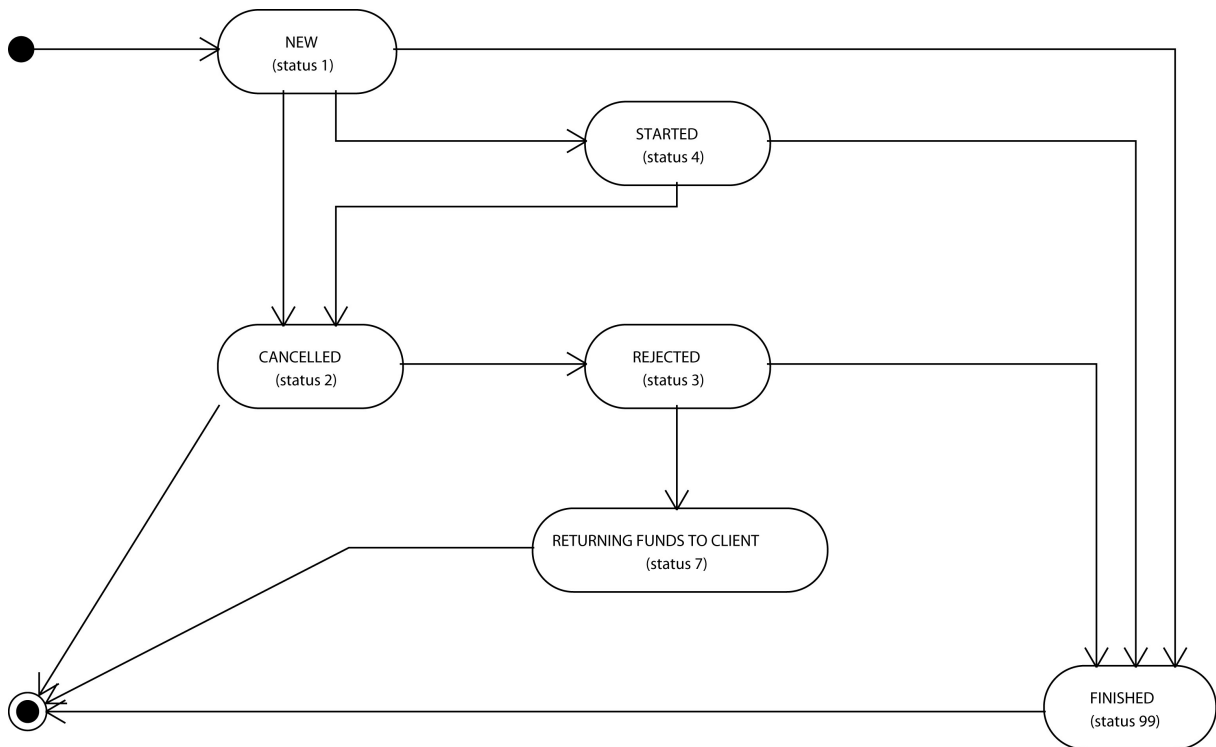
value	description
1	new
2	cancelled
3	rejected
4	started
5	awaiting collection
7	payment rejected, funds have been transferred by a client after the transaction was cancelled, or funds could not be returned automatically; such situations will be monitored and explained by the Płatności team
99	payment collected - finished
888	incorrect status - we ask user to contact us

2.2.1 Connections between particular stages of transaction

- if „Automatic payment collection” option is **disabled**:



- if „Automatic payment collection” option is **enabled**:



2.2.1 Additional information about transaction statuses

- Status 2 - “cancelled” will appear automatically a few days (defined in 2.4) after a transaction was created or started (Status 1 or 4) if it is not settled until then (funds will not be sent to Platnosci.pl system).
- Status 3 - “rejected” will appear when a User clicks on “Cancel” the moment the transaction is in status 5 “awaiting collection” and a selected payment method does not allow funds to be automatically returned to the client.
- Status 3 - “rejected” will also appear if a “cancelled” transaction (status 2) is settled (funds are transferred to Platnosci system). If a transaction of status 3 (“rejected”) is “collected” and if a User has not enabled the option of “Automatic payment collection”, its status will change to 5 - “awaiting collection”. Then, the User should click on “Receive” again to finish the transaction, i.e. go to status 99 - “finished”.
- Status 4 - “started” is a temporary stage and **does not have to take place**; the transaction may go to “awaiting collection” or “finished” status (if “Automatic payment collection” option is enabled) directly from “new” status.
- Status 5 - “awaiting collection” will only be displayed if the option „Automatic collection” is disabled, in such case the Shop has **5 days** (exactly 5 * 24 hours from the beginning of the transaction) to collect the payment. If such payment is not collected before the specified deadline, it will be **cancelled automatically**. The payment may be collected by invoking the Payment/confirm method or by using the website’s administration panel.
- Status 7 - “returning funds to client” will appear if transaction has status 3 “rejected” and the User clicks on “Cancel”.

2.3. Parameters of a new payment

parameter	required	data type	description
pos_id	yes	INT	value assigned by Platnosci.pl
pos_auth_key	yes	STR {7,7}	value assigned by Platnosci.pl
pay_type	no	ENUM	list of values presented by Platnosci.pl, point 2.4
session_id	yes	STR {1,1024}	payment id - unique for each client
amount	yes	NUM {1,10}	amount in grosz
desc	yes	STR {1,50}	short description - shown to a client, placed in statements and in other places
order_id	no	STR {1,1024}	order number
desc2	no	STR {0,1024}	any information
trsDesc	no	STR {0,27}	additional description for bank transfer
first_name	yes	STR {0,100}	name
last_name	yes	STR {0,100}	surname
street	no	STR {0,100}	street
street_hn	no	STR {0,10}	house number
street_an	no	STR {0,10}	flat number
city	no	STR {0,100}	city
post_code	no	STR {0,20}	postcode
country	no	STR {0,100}	client's country code (two letters) in accordance with ISO-3166 http://www.chemie.fu-berlin.de/diverse/doc/ISO_3166.html
email	yes	STR {0,100}	e-mail address
phone	no	STR {0,100}	phone number, you can enter several numbers separated with commas
language	no	ENUM	language code in accordance with ISO-639 http://www.ics.uci.edu/pub/ietf/http/related/iso639.txt (currently pl, en)
client_ip	yes	STR {7,15}	client's IP address in format: D{1,3}.D{1,3}.D{1,3}.D{1,3}
js	no	ENUM {0, 1}	value defines if the client's browser has Javascript enabled
payback_login	no	STR {0,40}	Client's PAYBACK login that will receive PAYBACK points
sig	no	STR {32}	checksum of form parameters being sent ¹
ts	no	STR	time tag used to calculate the sig value

Depending on the payment method, it may be necessary to enter the values defined in the table as not required. Additional information can be found next to descriptions of particular payment methods in point 2.4.

¹ More information: point 3.6

2.4. Payment methods

name	transaction values	time of automatic cancellation (in days) ²	description
m	0.50 - 999999.99	10	mTransfer - mBank
n	0.50 - 999999.99	10	MultiTransfer - MultiBank
w	0.50 - 999999.99	10	BZWBK - Przelew24
o	0.50 - 999999.99	10	Pekeo24Przelew - Bank Pekao
i	0.50 - 999999.99	10	Płać z Inteligo - Inteligo
d	0.50 - 999999.99	10	Płać z Nordea
p	0.50 - 999999.99	10	Płać z iPKO
h	0.50 - 999999.99	10	Płać z BPH
g	0.50 - 999999.99	10	Płać z ING
l	0.50 - 999999.99	10	LUKAS - e-przelew
u	0.50 - 999999.99	10	Eurobank
me	0.50 - 999999.99	10	Meritum Bank
wp	0.50 - 999999.99	10	Przelew z Polbank
wm	0.50 - 999999.99	10	Przelew z Millenium
wk	0.50 - 999999.99	10	Przelew z Kredyt Bank
wg	0.50 - 999999.99	10	Przelew z BGŻ
wd	0.50 - 999999.99	10	Przelew z Deutsche Bank
wr	0.50 - 999999.99	10	Przelew z Raiffeisen Bank
wc	0.50 - 999999.99	10	Przelew z Citibank
wn	0.50 - 999999.99	10	Przelew z Invest Bank
wi	0.50 - 999999.99	10	Przelew z Getin Bank
wy	0.50 - 999999.99	10	Przelew z Bankiem Pocztowym
c	1.01 - 7000.00 ³	5	credit card
b	0.50 - 999999.99	10	bank transfer
t	0.50 - 1000.00	1	test payment - a form is displayed where transaction status can be changed

The order of available payment channels in a shop should be as mentioned in this Document.

² Time after which the payment will be cancelled by the system if funds are not transferred to Płatności.pl system - it may change

³ It is possible to set limits individually.

2.4.1. Test - t

The test method is used for generating test payments. Funds from such transactions are not forwarded to the Shop; this method does not require any additional parameters for a new payment.

Test transactions are disabled by default, they are also automatically blocked 2 days after being used for the last time. In order to perform the tests, you need to activate this method of payment through “My shops” → “Shop name” → “List of payment points” → „Name of the point” option, then change the status for the “Test payment” by clicking on the “Status” column.

3. Integration with Platnosci.pl

3.1. Configuration data

In the Platnosci.pl application, each Shop can have many payment collecting points called POS.

The Shop specifies the following for each Pos: UrlPositive, UrlNegative and UrlOnline.

Platnosci.pl provides the Shop with the identifier of the created Pos, key1 and key2 string of characters (point 3.4). This data is available in Platnosci.pl administration panel after registering appropriate services.

You may find all the required configuration data in

“My shops” → “Shop name” → “List of payment points” → “Point name”.

3.2. Structure of the addresses: UrlPositive, UrlNegative

After a finished payment process, the Client can be redirected to an URL address specified by the Shop. Depending on the transaction status, an UrlPositive or UrlNegative address will be used accordingly. The Back to Shop addresses are of **informational** character only; no decisions can be taken on their basis.

The Back to Shop addresses can contain the following constants which will be exchanged for appropriate values, according to the table below:

constant	description
%transId%	identifier of a new transaction created in Platnosci.pl application
%posId%	pos_id values
%payType%	pay_type values
%sessionId%	session_id values
%amountPS%	amount values - dot as a separator
%amountCS%	amount values - comma as a separator
%orderId%	order_id values
%error%	error number in accordance to table 2.1, it is only used for UrlNegative

Examples:

`http://www.shop.pl=status_ok.html?pos id = %posId%&session id = %sessionId%`

`http://www.shop.pl=status_error.html?pos id = %posId%&session id = %sessionId%&error = %error%`

3.3. URL addresses of Platnosci.pl applications and available procedures

URL address for Platnosci.pl application is created in the following way:

$$URL = UrlPlatnosci.pl/Encoding/ProcedureName$$

where:

UrlPlatnosci.pl	Platnosci.pl application base address
Encoding	one of the following values: ISO, UTF, WIN
ProcedureName	one of the following values: NewPayment, Payment/get, Payment/confirm, Payment/cancel

3.3.1. Encoding

Depending on encoding page used by the Shop application, an appropriate encoding should be selected referring to Platnosci.pl procedures:

name in Platnosci.pl	applied encoding
ISO	ISO-8859-2
UTF	UTF-8
WIN	Windows-1250

3.3.2. Data format

For the following procedures: Payment/get, Payment/confirm, Payment/cancel, we can also specify the format used for sending data; thus following formula:

$$URL = UrlPlatnosci.pl/Encoding/ProcedureName/Format$$

where Format may be one of these values: „xml” or „txt”; “xml” is selected by default.

3.4. MD5 signatures

Each time a command is sent and each Platnosci.pl response generated, it is provided with an MD5 signature that allows to verify data integrity.

The signatures are created in accordance with the following formula (where “+” means operation of joining string of characters):

$$sig = md5(pos_id + session_id + value_1 + value_2 + ::: + value_n + ts + key)$$

where:

pos_id	value assigned by Platnosci.pl
session_id	payment identifier - unique for each client
value ₁ ... value _n	a list of additional values specified when describing particular methods
ts	any string of characters, recommended current time in seconds
key	string of characters known to Platnosci.pl and the Shop

In Platnosci.pl application, there are two key values assigned to a given pos_id:

key1 - key used while checking the signature sent by the Shop

key2 - key used to generate a signature that is sent to the Shop

3.5. Creating a new payment

To create a new payment, you need to place an appropriate form on your website which will redirect the Client to Platnosci.pl to NewPayment procedure (point 3.3). It is advised to use the POST method; if it is impossible, you may apply the GET method. The list of parameters and their description is available in point 2.3.

After the payment is finished, the Client will be redirected to UrlPositive or UrlNegative address using the GET method. It may happen that the Client does not go back to the Shop application, and that is why the information transferred using these addresses is not binding, decisions regarding payments cannot be made on its basis.

The method of transferring information about transactions is described in point 3.7.

An example of a form with a minimal amount of parameters:

```
<form action="https://www.platnosci.pl/paygw/ISO/NewPayment" method="POST" name="payform">
First name : <input type="text" name="first_name" value=""><br />
Last name : <input type="text" name="last_name" value=""><br />
Email : <input type="text" name="email" value=""><br />
<input type="hidden" name="pos_id" value="12345">
<input type="hidden" name="pos_auth_key" value="wq2i03q">
<input type="hidden" name="session_id" value="1234565">
<input type="hidden" name="amount" value="1000">
<input type="hidden" name="desc" value="Payment description">
<input type="hidden" name="client_ip" value="123.123.123.123">
<input type="hidden" name="js" value="0">
<input type="submit" value="Pay via Platnosci.pl">
</form>
<script language="JavaScript" type="text/javascript">
<!--
document.forms['payform'].js.value=1;
-->
</script>
```

3.6. Signing parameters transferred to a new payment

Optionally, the Shop application may add a checksum of all parameters transferred to the new payment form (NewPayment).

In order to use this option, we need to add two additional parameters to the form:

ts	time tag, value needed to verify a signature, any string of characters, e.g. time in seconds
sig	signature of transferred information

The sig value is calculated on the basis of the following formula:

```
sig = md5(pos_id + pay_type + session_id + pos_auth_key + amount + desc + desc2 + order_id +
first_name + last_name + payback_login + street + street_hn + street_an + city +
post_code + country + email + phone + language + client_ip + ts + key1)
```

If a given value is not transferred in a form creating a new payment we use a blank string of characters.

If the sig value is incorrectly calculated or the values of other transferred parameters are changed, the new payment will not be created. The client will be redirected to the UrlNegative address with error code 103.

3.7. Exchange of information regarding transactions

The Shop application is obliged to check signatures of transferred information.

3.7.1. Notifying the Shop about changing transaction status

Each change of the transaction status is reported to the Shop application. A POST request is sent to the given UrlOnline address, with the following parameters:

name	description
pos_id	Pos identifier
session_id	value given by the Shop while creating a payment
ts	time tag, value needed to verify signature
sig	signature of transferred information - point 3.4.

where the sig value is calculated according to the following formula:

$$sig = md5(pos_id + session_id + ts + key2)$$

When a message about the change of transaction status is sent, it does not contain any information. Transaction details and its current status **MUST** be read and analysed accordingly by the Shop application using the mechanisms described in point 3.7.2.

After receiving such a request, the Shop application **MUST** send an „OK” string in response. If another answer is received, it will be saved in the database and the notification will be considered not received.

The Shop application should account for situations where a notification is sent several times for the same transaction of the same status. For each repeated notification, an “OK” response should be sent as well.

For one Pos, one POST request is sent at the same time, but you should take into consideration the possibility to send a few requests to the same Pos at the same time.

Notifications are sent immediately after the payment status changes. If the notification is not collected by the Shop application, it will be sent again after the assigned period of time in accordance with the table below:

attempt	delay
0 - 10	1 minute
11 - 15	3 minutes
16 - 20	5 minutes
21 - 25	10 minutes
26 - 50	15 minutes
51 - 75	30 minutes
75 - 99	60 minutes
>= 100	stopped sending

3.7.2. Reading transaction status

In order to read the current transaction status, we need to invoke the Payment/get procedure (point 3.3.) using the POST method and entering the following parameters:

name	description
pos_id	pos identifier
session_id	transaction identifier
ts	time tag, value needed to verify signature
sig	signature of transferred information - point 3.4.

where the sig value is calculated according to the following formula:

$$sig = md5(pos_id + session_id + ts + key1)$$

In reply, we will receive the following pages containing information:

“txt” format:

```
status: OK
trans_id: 7
trans_pos_id: 1
trans_session_id: 417419
trans_order_id:
trans_amount: 200
trans_status: 5
trans_pay_type: t
trans_pay_gw_name: pt
trans_desc: Payment for test@test.pl
trans_desc2:
trans_create: 2004-08-23 10:39:52
trans_init: 2004-08-31 13:42:43
trans_sent: 2004-08-31 13:48:13
trans_rcv:
trans_cancel:
trans_auth_fraud: 0
trans_ts: 1094205761232
trans_sig: b6d68525f724a6d69fb1260874924759
```

„xml” format:

```
<?xml version="1.0" encoding="UTF-8" ?>
<response>
<status>OK</status>
<trans>
<id>7</id>
<pos_id>1</pos_id>
<session_id>417419</session_id>
<order_id></order_id>
<amount>200</amount>
<status>5</status>
<pay_type>t</pay_type>
<pay_gw_name>pt</pay_gw_name>
<desc>Payment for test@test.pl</desc>
<desc2></desc2>
<create>2004-08-23 10:39:52</create>
<init>2004-08-31 13:42:43</init>
<sent>2004-08-31 13:48:13</sent>
<recv></recv>
<cancel></cancel>
<auth_fraud>0</auth_fraud>
<ts>1094205828574</ts>
<sig>a95dc2145079b16a3668175279c35736</sig>
</trans>
</response>
```

In respect of data sent back by Platnosci.pl, we calculate the sig value according to the following formula:

$$sig = md5(pos_id + session_id + order\ id + status + amount + desc + ts + key2)$$

The description of particular notification fields is as follows:

Basic fields

txt field	xml field	description
status	response/status	notification processing status - for correct "OK"
trans_id	response/trans/id	unique transaction id assigned by Platinosci.pl
trans_pos_id	response/trans/pos_id	pos id for which transaction was created
trans_session_id	response/trans/session_id	value assigned by the Shop application when creating transaction
trans_order_id	response/trans/order_id	value assigned by the Shop application when creating transaction
trans_amount	response/trans/amount	current transaction value in grosz
trans_status	response/trans/status	current transaction status in accordance with point 2.2
trans_pay_type	response/trans/pay_type	payment type in accordance with point 2.4
trans_pay_gw_name	response/trans/pay_gw_name	gateway name performing transaction - internal information of Platinosci.pl application
trans_desc	response/trans/desc	value assigned by the Shop application when creating transaction
trans_desc2	response/trans/desc2	value assigned by the Shop application when creating transaction
trans_create	response/trans/create	transaction creation date
trans_init	response/trans/init	transaction starting date
trans_sent	response/trans/sent	date when transaction was transferred to be collected
trans_rcv	response/trans/rcv	transaction receiving date
trans_cancel	response/trans/cance	transaction cancelling date
trans_auth_fraud	response/trans/auth_fraud	internal information of Platinosci.pl application
trans_ts	response/trans/ts	value required to calculate signature
trans_sig	response/trans/sig	notification signature - result of md5 function

Additional fields - for selected payment methods:

- MultiBank, mBank, BZWBK

txt field	xml field	description
add_cc_number_hash	response/trans/add_cc_number hash	bank account number hash of payment sender (field filled in after some time)

- credit card

txt field	xml field	description
add_cc_number_hash	response/trans/add_cc_number hash	bank account number hash of payment sender
add_cc_bin	response/trans/add_cc_bin	BIN - bank id number of card issuer

- ING

txt field	xml field	description
add_cc_number_hash	response/trans/add_cc_number hash	bank account number hash of payment sender (field filled in after some time)
add_cc_number	response/trans/add_cc_number	bank account number of payment recipient
add_owner_name	response/trans/add_owner_name	payment recipient's name
add_owner_address	response/trans/add_owner_address	payment recipient's address
add_trans_title	response/trans/add_trans_title	payment title

- bank transfer

txt field	xml field	description
add_cc_number	response/trans/add_cc_number	bank account number of payment recipient
add_bank_name	response/trans/add_bank_name	payment recipient's bank name
add_owner_name	response/trans/add_owner_name	payment recipient's name
add_owner_address	response/trans/add_owner_address	payment recipient's address
add_trans_title	response/trans/add_trans_title	payment title
add_trans_prev	response/trans/add_trans_prev	link to page with bank transfer print preview

- test payment

txt field	xml field	description
add_test	response/trans/add_test	always "1" value
add_testid	response/trans/add_testid	transaction id

3.7.3. Receiving the payment

In order to receive a payment, i.e. confirm the transaction, we invoke the Payment/confirm procedure using the POST method, and enter the same parameters as in the case of reading transaction information (point 3.7.2).

3.7.4. Rejecting the payment

In order to cancel or reject the payment, we invoke the Payment/cancel procedure, and enter the same parameters as in the case of reading transaction information (point 3.7.2).

3.7.5. Operation performance status

In response to Payment/confirm and Payment/cancel procedures, we receive the following pages:

Correct performance - "txt" format:

```
status: OK
trans_id: 7
trans_pos_id: 1
trans_session_id: 417419
trans_ts: 1094206530505
trans_sig: 9da7c868407fedae6f1b6aca9054632b
```

Correct performance - "xml" format:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
<status>OK</status>
<trans>
<id>7</id>
<pos_id>1</pos_id>
<session_id>417419</session_id>
<ts>1094205828574</ts>
<sig>a95dc2145079b16a3668175279c35736</sig>
</trans>
</response>
```

In respect of data sent back by Platnosci.pl the sig value is calculated in accordance with the following formula:

$$sig = md5(pos_id + session_id + ts + key2)$$

Error - "txt" format:

```
status: ERROR
error_nr: 503
error_message:
```

Error - "xml" format:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
<status>ERROR</status>
<error>
<nr>503</nr>
<message></message>
</error>
</response>
```

3.8. Exchanging transaction information using WEBAPI/SOAP

A WEBAPI/SOAP interface can be used for described procedures. It can be found at:

<https://www.platnosci.pl/paygw/webapi/Payments>

An appropriate WSDL file can be downloaded from:

<https://www.platnosci.pl/paygw/webapi/Payments?wsdl>

If you have any questions, feel free to contact us.

4. Premium SMS

4.1. New transaction parameters for Premium SMS

parameter	required	data type	description
pos_id	yes	INT	value assigned by Platnosci.pl
pos_auth_key	yes	STR {7,7}	value assigned by Platnosci.pl
session_id	yes	STR {1,1024}	payment id - unique for each client
amount	yes	NUM {1,10}	amount in grosz
desc	yes	STR {1,50}	short description - shown to a client, placed in statements and in other places
order_id	no	STR {1,1024}	order number
desc2	no	STR {0,1024}	any information
first_name	no	STR {0,100}	name
last_name	no	STR {0,100}	surname
street	no	STR {0,100}	street
street_hn	no	STR {0,10}	house number
street_an	no	STR {0,10}	flat number
city	no	STR {0,100}	city
post_code	no	STR {0,20}	postcode
country	no	STR {0,100}	client's country code (two letters) in accordance with ISO-3166 http://www.chemie.fu-berlin.de/diverse/doc/ISO_3166.html
email	no	STR {0,100}	e-mail address
phone	no	STR {0,100}	phone number, you can enter several numbers separated with commas
language	no	ENUM	language code in accordance with ISO-639 http://www.ics.uci.edu/pub/ietf/http/related/iso639.txt (currently pl, en)
client_ip	yes	STR {7,15}	client's IP address in format: D{1,3}.D{1,3}.D{1,3}.D{1,3}
js	no	ENUM {0,1}	value defines if the client's browser has Javascript enabled

4.2. List of acceptable values for Premium SMS transaction

For Premium SMS transaction, it is possible to enter a transaction value in accordance with the table below:

value in grosz	Premium SMS number
122	71XX
244	72XX
366	73XX
610	75XX
732	76XX
1098	79XX

If any other value is entered, the transaction will not be created.

4.3 Creating new SMS transaction

In order to create a new SMS transaction, you have to place a special form on your page which will direct the Client to Platnosci.pl at:

UrlPlatnosci:pl/Encoding/NewSMS

where:

UrlPlatnosci.pl	Base address of Platnosci.pl application
Encoding	One of the following values: ISO, UTF, WIN

containing the rest of information required to activate the service by Premium SMS.

The form has to contain parameters as specified the table (point 4.1) - an example of the form below:

```
<form action="https://www.platnosci.pl/paygw/ISO/NewSMS" method="POST" name="smsform">
<input type="hidden" name="pos_id" value="12345">
<input type="hidden" name="pos_auth_key" value="w1Po3NB">
<input type="hidden" name="session_id" value="1234565">
<input type="hidden" name="amount" value="122">
<input type="hidden" name="desc" value="Service activation">
<input type="hidden" name="client_ip" value="123.123.123.123">
<input type="hidden" name="js" value="0">
<input type="submit" value="Activate the service by Premium SMS">
</form>
<script language="JavaScript" type="text/javascript">
<!--
document.forms['smsform'].js.value=1;
-->
</script>
```

4.4 Exchanging information about Premium SMS transactions

The Shop may read Premium SMS transaction status in the same way as for the other payment methods (point 3.7).

5. Website tools

5.1 Dynamic list of payment methods - JavaScript

A current payment list for a given Pos can be placed on a site by referring to JavaScript code loaded from Platnosci.pl.

The code can be found at:

URL = *UrlPlatnosci.pl/Encoding/js/PosId/K/paytype.js*

where:

UrlPlatnosci.pl	base address of Platnosci.pl application
Encoding	one of the following values: ISO, UTF, WIN
PosID	Pos id
K	two first signs from Key1 value

In paytype.js file there are the following methods:

PlnDrawSelect()	printing <select> element with available payment list
PlnDrawRadio()	printing radio elements list with names of payment methods
PlnDrawRadioImg(cols)	printing radio elements list with names and logotypes of payment methods; cols parameter - a number of columns which will be used to present a table

Example:

```
<script language='JavaScript' type='text/JavaScript'
src='https://www.platnosci.pl/paygw/ISO/js/1234/xx/paytype.js'>
</script>
<form action="https://www.platnosci.pl/paygw/ISO/NewPayment" method="POST" name="payform">
<input type="hidden" name="pos_id" value="12345">
<input type="hidden" name="session_id" value="1234565">
<input type="hidden" name="amount" value="1000">
<input type="hidden" name="desc" value="Payment description">
<script language='JavaScript' type='text/JavaScript'>
PlnDrawSelect();
</script>
<input type="hidden" name="client_ip" value="123.123.123.123">
<input type="hidden" name="js" value="0">
<input type="submit" value="Pay through Platnosci.pl">
</form>
<script language="JavaScript" type="text/javascript">
<!--
document.forms['payform'].js.value=1;
-->
</script>
```

5.2 Dynamic list of payment methods - xml

A list of the current payment methods for a given pos may be downloaded as xml file. An appropriate xml file can be found at:

URL = *UrlPlatnosci.pl/Encoding/xml/PosId/K/paytype.xml*

where:

UrlPlatnosci.pl	base address of Platnosci.pl application
Encoding	one of the following values: ISO, UTF, WIN
PosID	Pos id
K	two first signs from Key1 value

Example of a downloaded file:

```
<?xml version="1.0" encoding="UTF-8"?>
<paytypes>
<paytype>
<type>c</type>
<name>Payment card</name>
<enable>true</enable>
<img>https://www.platnosci.pl/paygw/images/paytype/on-c.gif</img>
<min>1.01</min>
<max>4000.0</max>
</paytype>
<paytype>
<type>m</type>
<name>mTransfer</name>
<enable>true</enable>
<img>https://www.platnosci.pl/paygw/images/paytype/on-m.gif</img>
<min>0.5</min>
<max>999999.99</max>
</paytype>
...
</paytypes>
```

5.3 Platnosci.pl basket

Platnosci.pl basket is a tool thanks to which you make it possible for a Buyer to purchase items online.

In order to use the service, you have to create a new basket in Platnosci.pl application and fill the required fields in. Then, so as the Buyer could add new items to it or display its content, you have to place special forms on your page which will direct to Platnosci.pl.

5.3.1 Adding items to basket

Here is the address of the page to which the Client should be directed:

UrlPlatnosci.pl/Encoding/cart/add

where:

UrlPlatnosci.pl	base address of Platnosci.pl application
Encoding	one of the following values: ISO, UTF, WIN

The form has to contain all the parameters mentioned in the table below:

parameter	data type	description
cart_id	INT	basket id assigned by Platnosci.pl
item_name	STR {1,128}	item name
price	NUM {10,2}	item price in PLN separated by “,” or “.”

Example of the form:

```
<form action="https://www.platnosci.pl/paygw/ISO/cart/add" method="POST" name="cartform">
<input type="hidden" name="cart_id" value="12345">
<input type="hidden" name="item_name" value="Book">
<input type="hidden" name="price" value="1,22">
<input type="submit" value="Add to basket">
</form>
```

5.3.2 Displaying basket content

This is a page to which the Client should be directed:

UrlPlatnosci.pl/Encoding/cart/display

where:

UrlPlatnosci.pl	base address of Platnosci.pl application
Encoding	one of the following values: ISO, UTF, WIN

The form has to contain a parameter mentioned in the table below:

parameter	data type	description
cart_id	INT	basket id assigned by Platnosci.pl

Example of the form:

```
<form action="https://www.platnosci.pl/paygw/ISO/cart/display" method="POST" name="cartform">  
<input type="hidden" name="cart_id" value="12345">  
<input type="submit" value="Show basket">  
</form>
```

6. Additional services

6.1 MassPay

MassPay is used to order many bank transfers for your clients.

6.1.1 File format with data for MassPay

In order to transfer a MassPay order, you have to prepare a file containing data required to perform this service. This is a text file where particular fields are separated by semi-colons - one line contains data for one order. Here is the explanation of the fields:

field number	format	description
1	text of 26 characters	full bank account number of the order beneficiary in NRM format - only digits without separation characters
2	number (separated by a dot or comma)	order amount
3	text - 35 characters at maximum	name and surname or company name of the beneficiary
4	text - 6 characters at maximum	beneficiary's postcode in the following format: DD-DDD
5	text	beneficiary's city
6	text	beneficiary's street name as well as house and flat number
7	text - 120 characters at maximum	order title

Empty lines as well as those which start with # in a file containing data are excluded. This file should be sent to Platosci.pl system using appropriate forms in the transaction system.

Note

- beneficiary's address, i.e. fields no. 4, 5, 6 cannot exceed 68 characters in total
- the following characters cannot be used in the data fields: ; * ' " ! + ? | < >

Example of a file:

```
121234567800000000001234567;1,50;Adam Kowalski;61-930;Poznań;ul. Nowa 3;Loyalty Program
121234567800000000001234567;1,99;Jan Nowak;60-930;Warszawa;ul. Marszałkowska 12;Remuneration
121234567800000000001234567;2,50;Adam Makowski;60-930;Poznań;ul. Marcelesińska 90/2;Promotion
```

7. Document changes history

version	date	author	description
1.55	2010-06-08	S. Jaranowski	- change of maximal transaction value for transactions realized via BZWBK – Przelew24 (p. 2.4)
1.54	2010-05-19	S. Jaranowski	- new payment methods (Eurobank, Meritum Bank, Invest Bank, Getin Bank, Bank Pocztowy) (p. 2.4)
1.53	2009-04-24	S. Jaranowski	- new parameter trsDesc (p. 2.3)
1.52	2009-01-15	S. Jaranowski	- parameters first_name, last_name and email were added to new payment form (p. 3.5)
1.51	2008-11-27	S. Jaranowski	- parameters first_name, last_name and email are required for all payment types (p. 2.3)
1.50	2008-11-25	S. Jaranowski	- new payment methods: Przelew z Polbank, Przelew z Millennium, Przelew z Kredyt Bank, Przelew z BGZ, Przelew z Deutsche Bank, Przelew z Raiffeisen Bank, Przelew z Citibank (p. 2.4)
1.49	2008-08-11	S. Jaranowski	- diagram of connections between transaction statuses (p. 2.2.1)
1.48	2008-06-10	S. Jaranowski	- additional fields describing transaction (p. 3.7.2)
1.47	2008-04-16	S. Jaranowski	- changed name of a payment method - from “Płać z PKO BP” to “Płać z iPKO” (p. 2.4) - new payment method: LUKAS e-przelew (p. 2.4) - deleted status 6 as unused
1.44, 1.45, 1.46	-	-	- internal changes - not published
1.43	2007-11-05	S. Jaranowski	- deleted duplicated text on Pekao bank in the table with a payment methods list (p. 2.4)
1.42	2007-10-15	S. Jaranowski	- change of order and arranging names of payment methods - amendments of contact data
1.41	2007-10-01	S. Jaranowski	- new payment method - Pekao S.A. (p. 2.4) - corrected value of maximum amount for credit cards - from 4000 to 7000 (p. 2.4) - document changes history added