GP webpay API HTTP

Technical specification for developers

Version: 1.0

Global Payments Europe, s.r.o.

Created **08.06.2016**

Last update **29.7.2016**

SERVICE. DRIVEN. COMMERCE

globalpaymentsinc.com

Author	GPE Product
Manager	GPE Application Development
Approved by	
Version	1.0
Confidentiality	Confidential

Document history:

Verze	Datum	Provedl	Komentář
0.1	08.06.2016	GPE Product	Initial document version – revision of the document GP_webpay_Seznameni_se_systemem_v2.1
0.2	13.06.2016	GPE Product	Corrections
1.0	17.06.2016	GPE Application Development	Document revision

Table of contents

1. Formula clause

This document including any possible annexes and links is intended solely for the needs of an e-shop service provider (hereinafter referred to as “Customer”).

Information included in this document (hereinafter referred to as "Information") are subject to intellectual property and copyright protection of the Global Payments Europe, s.r.o. (hereinafter referred to as "GPE") and are of a commercially confidential nature in accordance with the provisions of the section 504 of the Act No. 89/2012 Coll., Civil Code. The Customer is aware of the legal obligations in relation to the handling of Information.

Information or any part thereof may not be provided or in any way made available to third parties without the prior written consent of the GPE. At the same time, Information may not be used by the Customer for purposes other than for the purpose for which it serves. To avoid any doubts, without the prior written consent of the GPE, Information or any part thereof may be provided or in any way made available neither to companies providing payment processing services on the Internet.

The GPE to the extent permitted by applicable law retains all rights to this document and Information contained therein. Any reproduction, use, exposure, or other publication, or dissemination of Information or its part by methods known and as yet undiscovered without the prior written consent of the GPE is strictly prohibited. The GPE is not in any way responsible for any errors or omissions in Information. GPE reserves the right, without giving any reason, to amend or repeal any Information.

2. Introduction

Technical specification for developers “GP webpay API HTTP” aims at e-commerce developers of merchants (hereinafter referred to as the developer), who perform integration of the e-shop with the GP webpay payment gateway using the API HTTP.

Integration using the API WS is described in the technical specification for developers “GP webpay API WS”.

Important notice: it is the acquirer who enables individual payment methods and functionalities to the merchant. Information regarding ordering of the GP webpay payment gateway and contacts to all the acquirers are available on www.gpwebpay.cz.

3. Process of payment

3.1 Request

If the customer requires on-line payment, the merchant creates a request for creating a payment in his/her e-shop (hereinafter referred to as the request) and sends it to the GP webpay payment gateway interface API HTTP.

Request format for individual payment methods is described below. Complete list and sequence of parameters of a request are given in the following table:

Parameter	Type	Length	Mandatory
MERCHANTNUMBER **field included in digest**	character	10	yes
OPERATION **field included in digest**	character	20	yes
ORDERNUMBER **field included in digest**	numeric	15	yes
AMOUNT **field included in digest**	numeric	15	yes
CURRENCY **field included in digest**	numeric	3	yes/no *if not given, default currency from the merchant’s or bank’s settings is used*
DEPOSITFLAG **field included in digest**	numeric	1	yes
MERORDERNUM **field included in digest**	numeric	30	no
URL **field included in digest**	character	300	yes
DESCRIPTION **field included in digest**	character	255	no
MD **field included in digest**	character	255	yes/no
USERPARAM1 **field included in digest**	character	255	yes/no *mandatory for registration payment of the functionality Recurring*

			payment, otherwise not mandatory
FASTPAYID **field included in digest**	numeric	15	yes/no *mandatory if the Fastpay service is used*
PAYMETHOD **field included in digest**	character	255	no
DISABLEPAYMETHOD **field included in digest**	character	255	no
PAYMETHODS **field included in digest**	character	255	no
EMAIL **field included in digest**	character	255	no
REFERENCENUMBER **field included in digest**	character	20	no
ADDINFO **field included in digest**	XML scheme	24000	no
DIGEST	character	2000	yes
LANG **field NOT included in digest**	character	2	no

GP webpay API HTTP accepts only those requests, for which it can be proved that the originator of the request is an authorized subject, i.e. merchant with whom the acquirer has signed a contract.

DIGEST parameter is used to prove the origin of the request. Its content is generated on the basis of:

- Data sent: it proves that the contents of individual parameters has not been changed on the way to the system
- Private key: it proves that the request comes from the given merchant

When the integration begins, the merchant generates his/her private key using the GP webpay Portal; the merchant stores this key securely and provides it to the developer for integration. In the course of this process, the merchant's public key is stored automatically on the GP webpay server and before receiving a request from the merchant, it will be used to control if the merchant has signed the request with his/her private key.

DIGEST parameter, contained in the transmitted requests, contains electronic digest of all the other fields of the request. The digest ensures integrity and undeniableness of the transmitted request.

The request must meet the following conditions:

- In case that Redirect is used, the request is sent to the API HTTP by the GET method, or by means of sending the form data from the cardholder's internet browser by the GET or POST methods
- Parameters of the request must be signed in a clear and undeniable way. The DIGEST is created from the sent data contents using the merchant's private key (see the Annex no. 1 – Signing messages)

- Request is sent to the URL address according to the used environment:
    1. Client test environment: https://test.3dsecure.gpwebpay.com/pgw/order.do
    2. Production environment: https://3dsecure.gpwebpay.com/pgw/order.do
- Data transmitted in HTTP parameters of the request are x-www-form-urlencoded according to definition RFC 1866 – Chapter 8.2.2 (for more details see http://www.w3.org/MarkUp/html-spec/)
- HTTP request is sent via secured HTTPS channel using the server certificate provided by the GPE

In application GP webpay Portal, there can be downloaded other sources for integration with the GP webpay payment gateway using the API HTTP (e.g. examples of generating a digest (PHP, Java, .NET)).

After receiving the request, the GP webpay payment gateway creates an object named ORDER (see Chapter 4. Statuses of payment) and redirects the customer's browser to the payment page for payment method selection.

3.2 Response

After making the payment, the GP webpay payment gateway sends the result of payment to the merchant.

Response format for individual payment methods is described below.

All the responses from the GP webpay contain also the DIGEST fields, the content of which is generated:

- On the basis of data contained in the response
- And at the same time, on the basis of the GP webpay private key

When the integration begins, from the GP webpay Portal the merchant downloads the GPE public key, which serves to verify the content of the DIGEST field.

This way the merchant can verify that:

- The response really comes from the GP webpay
- The response has not been changed on the way.

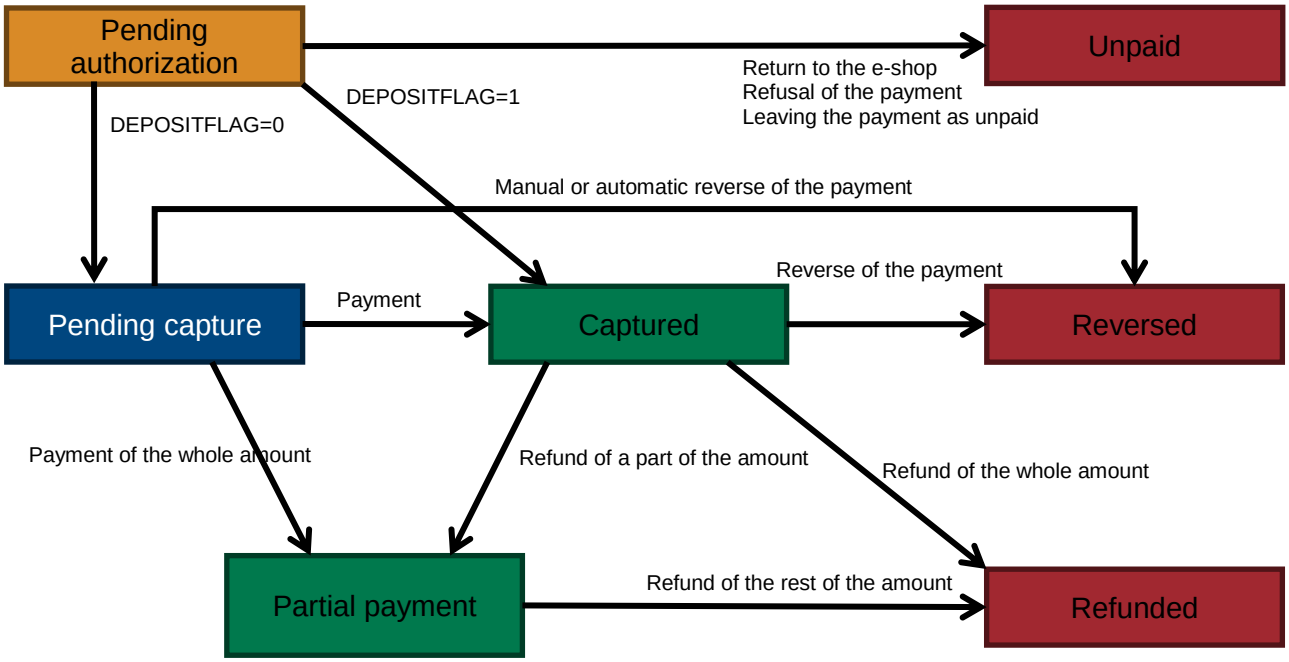
Important notice: when processing the response, it is necessary to use only the parameters that are sent back by the GP webpay payment gateway.

4. Statuses of payment

After receiving the request, the GP webpay payment gateway creates an object named ORDER. Further options of payment management depend on the status, in which the request (ORDER) is, see the table and status diagram:

Status of payment	Description of payment status

Captured	Payment has been captured. Payment will be credited to the e-shop's account according to the contract with the bank for card acceptance on the Internet.
Unpaid	Payment has not been captured. The reason can be non-completion of the payment by the customer on the GP webpay payment gateway, customer's return from the GP webpay payment gateway to the e-shop, decline of payment in the systems of GPE, card association, and issuer, or technical problem.
Refunded	Payment has been refunded. Refund has been made by the e-shop by means of the GP webpay Portal (menu "Payments"), or using the Web Services.
Partial payment	Payment has been paid partially or refunded partially. Partial payment has been made by the e-shop by means of the GP webpay Portal (menu "Payments"), or using the Web Services.
Pending capture	Payment has been authorized by the issuer and the paid amount has been blocked on the customer's account. E-shop has the option to capture the amount from the customer's account later by means of the GP webpay Portal (menu "Payments"), or using the Web Services.
Pending authorization	Payment is processed. E-shop has created a payment request and the customer has the option to pay on the GP webpay payment gateway. Standard payments can be paid until expiry of the time interval for payment, PUSH payments can be paid until expiry of the payment link or exhaustion of attempts for payment.
Reversed	Payment has been reversed. The reverse has been made by the e-shop by means of the GP webpay Portal (menu "Payments"), or using the Web Services, or the payment gateway GP webpay after expiry of the time interval for blocking the amount on the customer's account by the issuer.



5. Card payment

5.1 Request format

Parameter	Type	Length	Mandatory	Note
MERCHANTNUMBER	character	10	yes	A number assigned to each merchant.

field included in digest				
OPERATION **field included in digest**	character	20	yes	**CREATE_ORDER value**
ORDERNUMBER **field included in digest**	numeric	15	yes	**Ordinal number of the order. Every request from a merchant has to contain a unique order number.**
AMOUNT **field included in digest**	numeric	15	yes	The amount in the smallest units of the relevant currency For CZK = in hellers, for EUR = in cents
CURRENCY **field included in digest**	numeric	3	yes/no *if not given, default currency from the merchant's or bank's settings is used*	Currency identifier according to ISO 4217 (see Addendum ISO 4217). **Multicurrency (using of various currencies)** depends on support provided by the respective bank. It is necessary to address your bank in this respect.
DEPOSITFLAG **field included in digest**	numeric	1	yes	Specifies if the order has to be paid for automatically. Values allowed: 0 = instant payment not required 1 = payment required
MERORDERNUM **field included in digest**	numeric	30	no	Order identification for the merchant. *If not specified, the ORDERNUMBER value is used* *It is displayed in the bank statement.* ***Each bank has its own solution/limit – Addendum no. 3 – Maximal length of merchantOrderNumber field***
URL **field included in digest**	character	300	yes	Fully qualified merchant's URL. The request result is to be sent to this address. The result is resent via customer's browser – i.e. redirect is used (the GET method). *(including protocol specification - e.g. https://)* For security reasons, certain forms of URL address can be blocked – e.g. using of parameters in the address. This check cannot be switched off and it is necessary to test a real form of the return address in the testing environment.
DESCRIPTION **field included in digest**	character	255	no	Description of the purchase. The field content is transferred to the 3-D system for a later check by the card holder in the course of the authentication with the issuer's bank Access Control Server. The field may contain only ASCII characters ranging from 0x20 to 0x7E.
MD **field included in digest**	character	255	yes/no	Any merchant's data returned to the merchant in the response in the unchanged form – only "whitespace" characters are removed from both sides. The field is used to satisfy various demands of the e-shops. The field may only contain ASCII characters ranging from 0x20 to 0x7E. If it is necessary to transmit any other data, BASE64 encoding must be used (see Addendum no. 1 –

				BASE64 encoding and decoding). The field must not contain any personal data. **The resulting length of the data must not exceed 255 B.**
PAYMETHOD **field included in digest**	character	255	no	Value indicating the preferred payment method. Supported values: CRD – payment card MCM – MasterCard Mobile MPS – MasterPass
DISABLEPAYMETHOD **field included in digest**	character	255	no	Value indicating the **forbidden** payment method, even if it is enabled to the merchant. **It has higher priority than the "PAYMETHOD" field**. Supported values: CRD – payment card MCM – MasterCard Mobile MPS – MasterPass
PAYMETHODS **field included in digest**	character	255	no	List of allowed payment methods. Values are separated by comma ",". If the DISABLEPAYMETHOD field is defined at the same time, at first are found the same values and they are compared with the PAYMETHOD field. If the values differ, an error about an impropriate value in corresponding field is returned. Supported values: CRD – payment card MCM – MasterCard Mobile MPS – MasterPass
EMAIL **field included in digest**	character	255	no	Card holder's e-mail will be used for notification of the payment result and in the antifraud systems (FDS). The field must contain only **one** valid e-mail address. The field may contain any characters, but if e-mail address contains national characters, we recommend using see Addendum no. 1 – BASE64 encoding and decoding.
REFERENCENUMBER **field included in digest**	character	20	no	Internal ID at the merchant's Supported ASCII characters: x20(space), x23(#), x24($), x2A-x3B(*+,-./0-9:;), x3D(=), x40-x5A(@A-Z), x5E(^), x5F(_), x61-x7A(a-z)
ADDINFO **field included in digest**	XML scheme	24000	no	Basket description, data for FDS, additional information about the customer… May optionally be used for display the basket in wallets (MasterPass). **We highly recommend sending requests to the payment gateway using the POST method. This removes the limit of data length in the address bar (GET method) and ensures preservation of the national characters coding in UTF-8 format.** **Another recommendation is not to use spacing and spaces/whitespaces between XML elements. Browsers usually do not work very correctly with it and interpret spacing differently. In most cases this ends with signature non-verification on the server.**
DIGEST	character	2000	yes	A check signature of the string generated as a concatenation of the fields in the order given in this table – Annex no. 1 – Signing messages

				In case of the incorrect data signature the exception report is sent back to the Internet browser, which has sent this request.
LANG **field NOT included in digest**	character	2	no	Value indicating automatic choice of language at the payment gateway. Abbreviation of one of the supported languages must be used – see the list at the payment gateway.

5.2 Response format

Parameter	Type	Length	Mandatory	Note
OPERATION **field included in digest**	character		yes	**CREATE_ORDER value**
ORDERNUMBER **field included in digest**	numeric	15	yes	Contents of the field from the request.
MERORDERNUM **field included in digest**	numeric	30	no	Contents of the field from the request, if included.
MD **field included in digest**	character	255	no	Contents of the field from the request, if included.
PRCODE **field included in digest**	numeric		yes	Primary code. For details, see “List of return codes”.
SRCODE **field included in digest**	numeric		yes	Secondary code. For details, see “List of return codes”.
RESULTTEXT **field included in digest**	character	255	no	A text description of the error identified by a combination of PRCODE and SRCODE. The contents are coded using the Windows Central European (Code Page 1250).
USERPARAM1 **field included in digest**	character	64	yes/no *only if the merchant has this functionality enabled*	Hash numbers of the payment card. Hash is a unique value for each and every card and merchant – i.e. if the payment is made by the same card at the same merchant, the resulting hash is identical, if the same card is used at another merchant, there is another hash.
ADDINFO **field included in digest**	XML scheme		no	The field is filled in depending on settings of the input parameters for wallets (MasterPass) and requested return information (payment card brand…). **If sending this field is requested (depends on data settings in the “ADDINFO” input parameter), response will be sent by POST method. The reason is the size limit of data sent by the GET method (address barcode of the browser) and secure determination of character set of the response – UTF-8.**
DIGEST	character	2000	yes	A check signature of the string generated as a concatenation of all the fields sent in the given order – Annex no. 1 – Signing messages
DIGEST1	character	2000	yes	A check signature of the string generated as a concatenation of all the fields sent in the given order (without the DIGEST field) and on the top of that also the MERCHANTNUMBER field (the field is not sent, the merchant has to know it, the field is added to the end of the string). Security and unambiguity of the response is increased in this way.

				Verification of the signature is identical to the DIGEST field.

The merchant must work ONLY with fields that he/she RECEIVES, not with fields about which he/she "thinks" that should be received.

6. Payment using digital wallet

6.1 MasterPass

GP webpay API HTTP offers the following options:

- Creating a payment and sending the shopping basket, which is displayed in the wallet
- Dividing a payment into two steps:
    1. Creating the order in a standard way and getting response on what type of card will be used for payment
    2. Payment confirmation (it is possible to modify the amount) and the payment completion

To send the basket, the standard ADDINFO field is used. In this field, there are saved data in the XML format.

Order parameters are identical as in case of a standard order, but in addition to it, it is necessary to set "requestDeferredAuthorization" element to value "true" in the ADDINFO parameter (to get the address, it is necessary to set "requestShippingDetails"element to true, to get loyalty programme it is necessary to set "requestLoyaltyProgram" element" to true). Due to this setting the payment process is interrupted and after all the data is received from the MasterPass environment, further processing is redirected to the merchant's URL, which was given at order creation. Response format is identical/simplified and contains following parameters: PRCODE = 200, SRCODE = 0. In the ADDINFO field (in xml), there are information about the card holder and the merchant can work with them.

The merchant processes the data received and by calling the standard interface he/she can modify input parameter of the original order.

To fully use the potential offered by MasterPass, the MasterPass service can be offered directly on the web pages of the e-shop by means of the "Buy with MasterPass" button. Possibilities of integration of the e-shop with MasterPass are described in the technical specification for developers "GP webpay MasterPass Integration manual", which is sent on request by the GPE Application Support.

6.1.1 Request format

Parameter	Type	Length	Mandatory	Note
MERCHANTNUMBER **field included in digest**	character	10	yes	A number assigned to each merchant.
OPERATION **field included in digest**	character	20	yes	**FINALIZE_ORDER value**

Parameter	Type	Length	Mandatory	Note
ORDERNUMBER **field included in digest**	numeric	15	yes	**Order number – must correspond with the original order number**
AMOUNT **field included in digest**	numeric	15	yes	The amount in the smallest units of the relevant currency For CZK = in hellers, for EUR = in cents
URL **field included in digest**	character	300	yes	Fully qualified merchant's URL. The request result is to be sent to this address. The result is resent via customer's browser – i.e. redirect is used (the GET method). *(including protocol specification - e.g. https://)* For security reasons, certain forms of URL address can be blocked – e.g. using of parameters in the address. This check cannot be switched off and it is necessary to test a real form of the return address in the testing environment.
DIGEST	character	2000	yes	A check signature of the string generated as a concatenation of the fields in the order given in this table. *In case of the incorrect data signature the exception report is sent back to the Internet browser, which has sent this request.*

6.1.2 Response format

Parameter	Type	Length	Mandatory	Note
OPERATION **field included in digest**	character	20	yes	**FINALIZE_ORDER value**
ORDERNUMBER **field included in digest**	numeric	15	yes	Contents of the field from the request.
MERORDERNUM **field included in digest**	numeric	30	no	Contents of the field from the operation CREATE_ORDER, if included.
MD **field included in digest**	character	255	no	Contents of the field from the operation CREATE_ORDER, if included and not empty.
PRCODE **field included in digest**	numeric		yes	Primary code. For details, see "List of .return codes"
SRCODE **field included in digest**	numeric		yes	Secondary code. For details, see "List of return codes".
RESULTTEXT **field included in digest**	character	255	no	A text description of the error identified by a combination of PRCODE and SRCODE. The text is sent without diacritic.
USERPARAM1 **field included in digest**	character	64	yes/no *only if the merchant has enabled this functionality*	Hash numbers of the payment card. Hash is a unique value for each and every card and merchant – i.e. if the payment is made by the same card at the same merchant, the resulting hash is identical, if the same card is used at another merchant, there is another hash
ADDINFO **field included in digest**	XML scheme		no	The field is filled in depending on settings of the input parameters for wallets (MasterPass) and requested return information (payment card brand…).

				If sending this field is requested (depends on data settings in the "ADDINFO" input parameter), response will be sent by POST method. The reason is the size limit of data sent by the GET method (address barcode of the browser) and secure determination of character set of the response – UTF-8.
DIGEST	character	2000	yes	A check signature of the string generated as a concatenation of all the fields sent in the given order.
DIGEST1	character	2000	yes	A check signature of the string generated as a concatenation of all the fields sent in the given order (without the DIGEST field) and on the top of that also the MERCHANTNUMBER field (the field is not sent, the merchant has to know it, the field is added to the end of the string). Security and unambiguity of the response is increased in this way. *Verification of the signature is identical to the DIGEST field.*

7. Payments with payment button

7.1 PLATBA 24

PLATBA 24 can be offered directly on the webpages of the e-shop by means of the "PLATBA 24" button. To integrate e-shop for this case of use, the "PAYMETHOD" parameter with value "BTNCS" is used in the request:

Parameter	Type	Length	Mandatory	Note
PAYMETHOD **field included in digest**	character	255	no	Value indicating the preferred payment method. Supported values: CRD – payment card MCM – MasterCard Mobile MPS – MasterPass **BTNCS – PLATBA 24 – payment button České spořitelny**

8. Payments facilitating functionalities

8.1 Recurring payment

8.1.1 Registration payment

The first one, the so-called registration payment, is made as a standard payment 3D Secure and the card holder has to be verified in that and the payment has to be made. Then the recurring payment can be created.

Registration payment is marked by adding the "USERPARAM1" parameter to the request:

Parameter	Type	Length	Mandatory	Note
USERPARAM1 **field included in digest**	character	255	yes/no *mandatory for*	User's field. Now used for submission of "R" parameter –

Parameter	Type	Length	Mandatory	Note
			registration of the "master" payment, otherwise not compulsory	information about a request for registration of "master" recurring payment.

This parameter is located behind the MD parameter – see the list/sequence of fields.

Response format is identical to a standard format.

8.1.2 Recurring payment

Recurring payment is made using the API WS (Web Services) without redirecting of the customer's browser to the payment page for entering payment card data (see the technical specification for developers "GP webpay API WS").

8.2 Fastpay

Fastpay feature enables the merchant to display on the payment page for the logged in customer last 4 digits of the payment card and the card validity of the card, which the customer has used for the previous payment.

To integrate e-shop for this case of use, the "FASTPAYID" parameter with value "ORDERNUMBER" from the previous payment is used in the request:

Parameter	Type	Length	Mandatory	Note
FASTPAYID **field included in digest**	numeric	15	yes/no *mandatory if the Fastpay service is used*	A unique ORDERNUMBER of the order, which was used in the past and should serve as a basis to pre-fill card number. The order should be paid and cannot be older than 12 (18) months, as it may have been automatically removed from the system.

If the relevant payment is not found, data are not displayed.

This parameter is located behind the MD parameter – see the list/sequence of fields.

Response format is identical to a standard format.

9. Annexes and addenda

9.1 Annex no. 1 – Signing messages

9.1.1 Signing a request

GP webpay accepts only requests for which it can be proved that the originator of the request is an authorized subject (i.e. merchant) with whom GPE, s.r.o. has signed a contract for GP webpay services.

The DIGEST field is used to prove the origin of the request. Its contents are generated based on the following data:

- Data sent – this data is used to prove that the contents of the fields have not been changed on the way to the system.
- Private key – the private key is used to prove that the request comes from the merchant.

At the moment of beginning the integration, the merchant using the GP webpay Portal generates a private key, which he/she stores securely and provides it to the developer for integration. The merchant's public key is stored automatically on the GP webpay server and before the merchant,s request is accepted, it will be used for verifying, if the merchant has signed the request by his/her private key.

DIGEST parameter contained in transmitted requests contains electronic signature of all other fields of the request. The electronic signature guarantees integrity and undeniableness of the transmitted request.

Any request not containing the DIGEST field or with non-matching contents of the DIGEST field will be rejected with the following explanation:

- PRCODE=5 SRCODE=34 "Mandatory field missing, DIGEST" or
- PRCODE =31 "Invalid signature".

To generate and verify the electronic signature, a string composed as a concatenation of the text interpretation of the values of all fields contained in the request sent, except from the DIGEST field. When compiling the input message, the merchant has to use the same order of fields as that used in the definition of the request and intersperse individual fields by delimiter "|" (pipe, ASCII 124, hexa 7C). The delimiter must not be preceded or followed by whitespace. URLEncode parameters are used only for data transmission, original data have to be used to generate a signature.

Source for generating the DIGEST field in case of method CREATE_ORDER is the value created by concatenation of the fields in the order given here:

MERCHANTNUMBER + | + OPERATION + | + ORDERNUMBER + | + AMOUNT + | + CURRENCY + | + DEPOSITFLAG + | + MERORDERNUM + | + URL + | + DESCRIPTION + | + MD

If the request does not contain any of optional fields, this field is skipped. If the field is sent empty, it is necessary to include it in generating for the DIGEST field and in the string, there will be two separators next to each other – ||.

If the merchant sends only obligatory parameters, for generating the DIGEST field serves the value: MERCHANTNUMBER + | + OPERATION + | + ORDERNUMBER + | + AMOUNT + | + CURRENCY + | + DEPOSITFLAG + | + URL

9.1.2 Response verification

All the responses from the GP webpay system also contain the DIGEST field. Its contents are generated as follows:

- based on the data contained in the response;
- and, at the same time based on the GP webpay private key.

At the moment of beginning the integration, the merchant downloads the GP webpay public key from the GP webpay Portal. It is used by the merchant to verify the contents of the DIGEST field.

This way the merchant can easily verify that:

- the response really comes from the GP webpay;
- the response has not been changed on the way to the merchant.

Furthermore, the response contains also the DIGEST1 parameter, which further enhances the security of the response. The DIGEST1 parameter is generated as the DIGEST parameter, but parameter "MERCHANTNUMBER" is added to the parameters for validation of the DIGEST parameter. This parameter is not sent in the response and the merchant has to add it by himself/herself because he/she knows its value.

The resulting string for validation of the DIGEST1 field looks like this:

<string for the field DIGEST> + | + MERCHANTNUMBER

9.1.3 Generating the electronic signature

Inputs:

- Data message (message)
- Private RSA key (with a K-length modulus)

Outputs:

- Electronic signature (BASE64 encoded), approximate length K*1.5

The electronic signature is generated as follows

a) the value of the function SHA-1 [3] is derived from the message

b) the hash is encoded into the input value for the RSA signature, using the EMSA-PKCS1-v1_5-ENCODE algorithm as described in paragraph 9.2.1 [1]. The encoding is made as follows:

01 | FF* | 00 | 30 21 30 09 06 05 2B 0E 03 02 1A 05 00 04 14 | hash

where FF characters are repeated as many times as necessary for the total length of the string to be one octet shorter than the key modulus. The character | is used for the strings concatenation.

c) the RSA signature is calculated using the output value from b), as described in 8.1.1 [1] RSASSA-PKCS1-V1_5-SIGN

d) The output from c) is encoded using BASE64

9.1.4 Verification of the electronic signature

Inputs:

- Data message
- Electronic signature (BASE64 encoded)
- Public RSA key

Outputs:

- Logical value - YES – the signature is valid
- Logical value - NO – the signature is invalid or its verification has not been possible.

The electronic signature is verified as described in 8.1.2 [1] in the following main steps:

a) depending on the settings for the merchant in the GPE system, the correct public key is selected and its integrity is verified;

b) the electronic signature is decoded using BASE64;

c) the output from b) is decrypted using the selected public key;

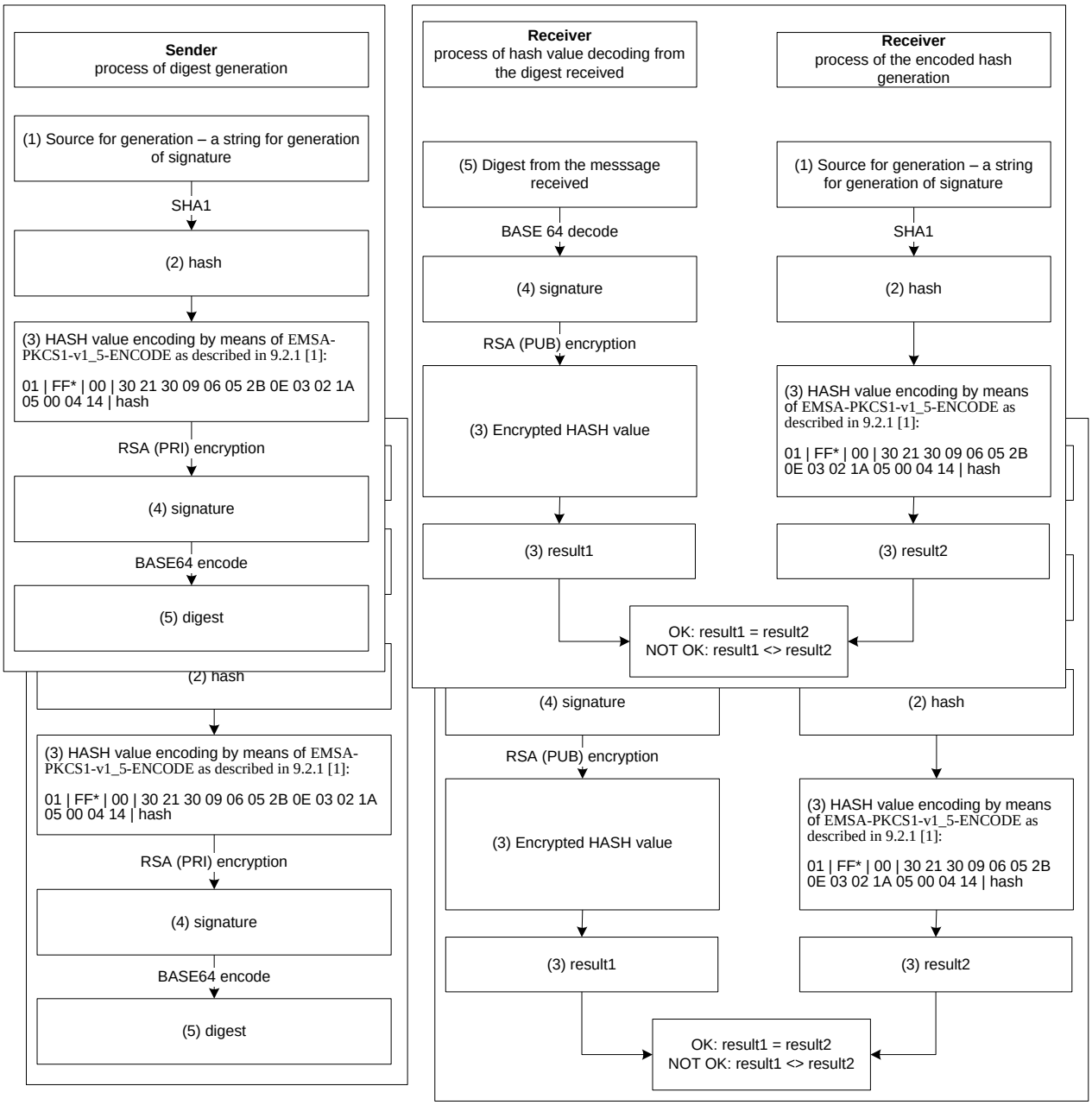
d) a miniature (hash) is generated based on the message and encoded as described in “Generating of the electronic signature”, paras a) and b);

e) the electronic signature decoded according to c) is compared with the result from d). If they are identical, the function returns a logical truth (the signature is valid).

Otherwise, the function returns a logical untruth (the signature is not valid).

The application used for verification of the electronic signature has to identify a signature as invalid also in the case, if verification of the signature has not been possible (for example, due to unavailability of the key).

9.1.5 Graphic representation of key generation and verification



9.1.6 Keys used

To generate the electronic signature (DIGEST), RSA keys (keyPair) are used with a modulus length of 2048 bits. During the communication between GP webpay and the merchant, the following key pairs are used:

GPE's KeyPair	GPE's private key (GPE_{PRI})	Used for the calculation of the electronic signature for messages sent by GPE.	
	GPE's public key (certificate) (GPE_{PUB})	Used by the merchant to verify the electronic signature in messages sent by GPE.	Delivered in the form of a X509 certificate
Merchant's KeyPair	Merchant's private key ($MERCH_{PRI}$)	Used for generating the electronic signature for messages sent by the merchant.	

	Merchant's public key (certificate) ($MERCH_{PUB}$)	Used by GPE to verify the electronic signature in messages sent by the merchant.	Delivered in the form of a X509 self-signed certificate

The application used to generate a self-signed certificate is delivered to the merchant when the merchant applies GPE, s.r.o. for signing a contract. Commercially issued keys can be used as well, but their validity is limited to 1 or 2 years (in comparison with the key generated by the application, there the key validity is longer).

9.1.7 Logging

The application used to verify the electronic signature must store in its audit logs all information about successful and non-successful verification of the electronic signature.

For the purpose of verification of the audit logs, all data required for the verification and re-verification of the electronic signature must be logged. This data includes mainly the electronic signature, the fields, which have been used for its generation, and the result of its verification. If any logs are missing or incomplete, the authenticity of such transactions cannot be confirmed.

9.1.8 References

For further information about the mechanism used to generate the DIGEST field, see the following documents:

[1] RFC 2437, PKCS #1: RSA Cryptography Specifications, October 1998;

[2] XML-Signature Syntax and Processing, W3C Recommendation 12 February 2002,

http://www.w3.org/TR/xmldsig-core/;

[3] RFC 3174 - US Secure Hash Algorithm 1 (SHA1), September 2001;

[4] RFC 2459 – Internet X.509 Public Key Infrastructure Certificate and CRL Profile,

January 1999

The following cryptographic libraries and components may be used to generate the electronic signature:

- JCE Cryptix: Alternative JCE provider offering an algorithm for the RSA/SHA1/PKCS#1 signature, **www.cryptix.org**
- Bouncy Castle: Alternative JCA provider offering libraries for the generation of certificates and work with the PKCS#12 certificate storage, www.bouncycastle.org.
- Crypto++, a free C++ class library of cryptographic schemes supporting also the RSA/SHA1/PKCS#1 algorithm, www.cryptopp.com

9.2 Annex no. 2 – List of return codes

The result of the processing of the request in GP webpay is described as a pair of return codes. If these return codes are different from zero PRCODE describes the type of error. If SRCODE is different from zero it describes the error in detail.

Example:

PRCODE=1 SRCODE=8 means that the DEPOSITFLAG field in the request received has been too long. The RESULTTEXT code returned in this case is “Field too long, DEPOSITFLAG”.

9.2.1 PRCODE / primaryReturnCode

Value	Meaning in Czech	Meaning in English
0	OK	OK
1	Pole příliš dlouhé	Field too long
2	Pole příliš krátké	Field too short
3	Chybný obsah pole	Incorrect content of field
4	Pole je prázdné	Field is null
5	Chybí povinné pole	Missing required field
11	Neznámý obchodník	Unknown merchant
14	Duplikátní číslo objednávky	Duplicate order number
15	Objekt nenalezen	Object not found
17	Částka k úhradě překročila autorizovanou částku	Amount to deposit exceeds approved amount
18	Součet kreditovaných částek překročil uhrazenou částku	Total sum of credited amounts exceeded deposited amount
20	Objekt není ve stavu odpovídajícím této operaci *Info: Pokud v případě vytváření objednávky (CREATE_ORDER) obdrží obchodník tento návratový kód, vytvoření objednávky již proběhlo a objednávka je v určitém stavu – tento návratový kód je zapříčiněn aktivitou držitele karty (například pokusem o přechod zpět, použití refresh…).*	Object not in valid state for operation
25	Uživatel není oprávněn k provedení operace	Operation not allowed for user
26	Technický problém při spojení s autorizačním centrem	Technical problem in connection to authorization centre
27	Chybný typ objednávky	Incorrect order type
28	*Zamítnuto v 3D* *Info: důvod zamítnutí udává SRCODE*	Declined in 3D
30	*Zamítnuto v autorizačním centru* *Info: Důvod zamítnutí udává SRCODE*	Declined in AC
31	Chybný podpis	Wrong digest
35	Expirovaná session Nastává při vypršení webové session při zadávání karty	Session expired
50	Držitel karty zrušil platbu	The cardholder cancelled the payment

200	Žádost o doplňující informace	Additional info request
1000	Technický problém	Technical problem

9.2.2 SRCODE / secondaryReturnCode

Value	Meaning in Czech	Meaning in English
0	Bez významu	No meaning
If PRCODE is 1 to 5, 15 and 20, the following SRCODE may return		
1	ORDERNUMBER	ORDERNUMBER
2	MERCHANTNUMBER	MERCHANTNUMBER
6	AMOUNT	AMOUNT
7	CURRENCY	CURRENCY
8	DEPOSITFLAG	DEPOSITFLAG
10	MERORDERNUM	MERORDERNUM
11	CREDITNUMBER	CREDITNUMBER
12	OPERATION	OPERATION
18	BATCH	BATCH
22	ORDER	ORDER
24	URL	URL
25	MD	MD
26	DESC	DESC
34	DIGEST	DIGEST
If PRCODE is 28, the following SRCODE may return		
3000	**Neověřeno v 3D. Vydavatel karty není zapojen do 3D nebo karta nebyla aktivována.** *Info: Ověření držitele karty bylo neúspěšné (neplatně zadané údaje, stornování autentikace, uzavření okna pro autentikaci držitele karty se zpětnou vazbou…).* *V transakci se nesmí pokračovat.*	**Declined in 3D. Cardholder not authenticated in 3D.** *Note: Cardholder authentication failed (wrong password, transaction cancelled, authentication window was closed…).* *Transaction Declined.*
3001	**Držitel karty ověřen.** *Info: Ověření držitele karty v 3D systémech proběhlo úspěšně. Pokračuje se autorizací objednávky.*	**Authenticated** *Note: Cardholder was successfully authenticated – transaction continue with authorization.*
3002	**Neověřeno v 3D. Vydavatel karty nebo karta není zapojena do 3D.** *Info: V 3D systémech nebylo možné ověřit držitele karty – karta, nebo její vydavatel, není zapojen do 3D.* *V transakci se pokračuje.*	**Not Authenticated in 3D. Issuer or Cardholder not participating in 3D.** *Note: Cardholder wasn't authenticated – Issuer or Cardholder not participating in 3D.* *Transaction can continue.*
3004	**Neověřeno v 3D. Vydavatel karty není zapojen do 3D nebo karta nebyla aktivována.** *Info: V 3D systémech nebylo možné ověřit držitele karty – karta není aktivována, nebo její vydavatel, není zapojen do 3D.* *V transakci je možné pokračovat.*	**Not Authenticated in 3D. Issuer not participating or Cardholder not enrolled.** *Note: Cardholder wasn't authenticated – Cardholder not enrolled or Issuer or not participating in 3D.* *Transaction can continue.*
3005	**Zamítnuto v 3D.Technický problém při ověření držitele karty.**	**Declined in 3D. Technical problem during Cardholder authentication.**

	Info: V 3D systémech nebylo možné ověřit držitele karty – vydavatel karty nepodporuje 3D, nebo technický problém v komunikaci s 3D systémy finančních asociací, či vydavatele karty. *V transakci není možné pokračovat, povoleno z důvodu zabezpečení obchodníka před případnou reklamací transakce držitelem karty.*	*Note: Cardholder authentication unavailable – issuer not supporting 3D or technical problem in communication between associations and Issuer 3D systems.* *Transaction cannot continue.*
3006	**Zamítnuto v 3D. Technický problém při ověření držitele karty.** *Info: V 3D systémech nebylo možné ověřit držitele karty – technický problém ověření obchodníka v 3D systémech, anebo v komunikaci s 3D systémy finančních asociací, či vydavatele karty.* *V transakci není možné pokračovat.*	**Declined in 3D. Technical problem during Cardholder authentication.** *Note: Technical problem during cardholder authentication – merchant authentication failed or technical problem in communication between association and acquirer.* *Transaction cannot continue.*
3007	**Zamítnuto v 3D. Technický problém v systému zúčtující banky. Kontaktujte obchodníka.** *Info: V 3D systémech nebylo možné ověřit držitele karty – technický problém v 3D systémech.* *V transakci není možné pokračovat.*	**Declined in 3D. Acquirer technical problem. Contact the merchant.** *Note: Technical problem during cardholder authentication – 3D systems technical problem.* *Transaction cannot continue.*
3008	**Zamítnuto v 3D. Použit nepodporovaný karetní produkt**. *Info: Byla použita karta, která není v 3D systémech podporována.* *V transakci není možné pokračovat.*	**Declined in 3D. Unsupported card product.** *Note: Card not supported in 3D.* *Transaction cannot continue.*
If PRCODE is 30, the following SRCODE may return		
1001	**Zamitnuto v autorizacnim centru, karta blokovana**[1] *Zahrnuje důvody, které naznačují zneužití platební karty – kradená karta, podezření na podvod, ztracená karta apod.* *Většinou pokus o podvodnou transakci.*	**Declined in AC, Card blocked** *Includes the reasons implying that the card has been misused – stolen card, suspected card fraud, lost card, etc.*
1002	**Zamitnuto v autorizacnim centru, autorizace zamítnuta** *Z autorizace se vrátil důvod zamítnutí "Do not honor".* *Vydavatel, nebo finanční asociace zamítla autorizaci BEZ udání důvodu.*	**Declined in AC, Declined** *Reason:* *Card Issuer or financial association rejected authorization (Do Not Honor)*
1003	**Zamitnuto v autorizacnim centru, problem karty** *Zahrnuje důvody:* *expirovaná karta, chybné číslo karty, nastavení karty - pro kartu není povoleno použití na internetu, nepovolená karta, expirovaná karta, neplatná karta, neplatné číslo karty, částka přesahuje maximální limit karty, neplatné CVC/CVV, neplatná délka čísla karty, neplatná expirační doba, pro kartu je požadována kontrola PIN.*	**Declined in AC, Card problem** *Possible reasons:* *Expired card, wrong card number, Internet transaction not permitted to Cardholder, invalid card, invalid card number, amount over card maximum limit, wrong CVC/CVV, invalid card number length, invalid expiry date, PIN control is required for used card*
1004	**Zamitnuto v autorizacnim centru, technicky problem** *Autorizaci není možné provést z technických důvodů – technické problémy v systému vydavatele karty, nebo finančních asociací a finančních procesorů.*	**Declined in AC, Technical problem in authorization process** *Authorization rejected – technical problem* *Technical problem in card Issuer systems or financial associations systems (Card Issuer unavailable)*
1005	**Zamitnuto v autorizacnim centru, Problem uctu** *Důvody: nedostatek prostředků na účtu, překročeny limity, překročen max. povolený počet použití…*	**Declined in AC, Account problem** *Possible reasons: finance absence, over account limit, over daily limit*

[1] Only the bold part in this and the following cells of this column will be included in the RESULTTEXT field (optional field) in a response sent to the merchant. Other text is only the explanation for merchants.

If authorization is rejected, the payment gateway receives the return code directly from the card issuer (or from the service provider, or financial association). If the rejected authorization is claimed, the cardholder has to contact his card issuing bank, which responses him directly, or this bank resolves a claim with the bank, which processed the transaction (merchant's bank).

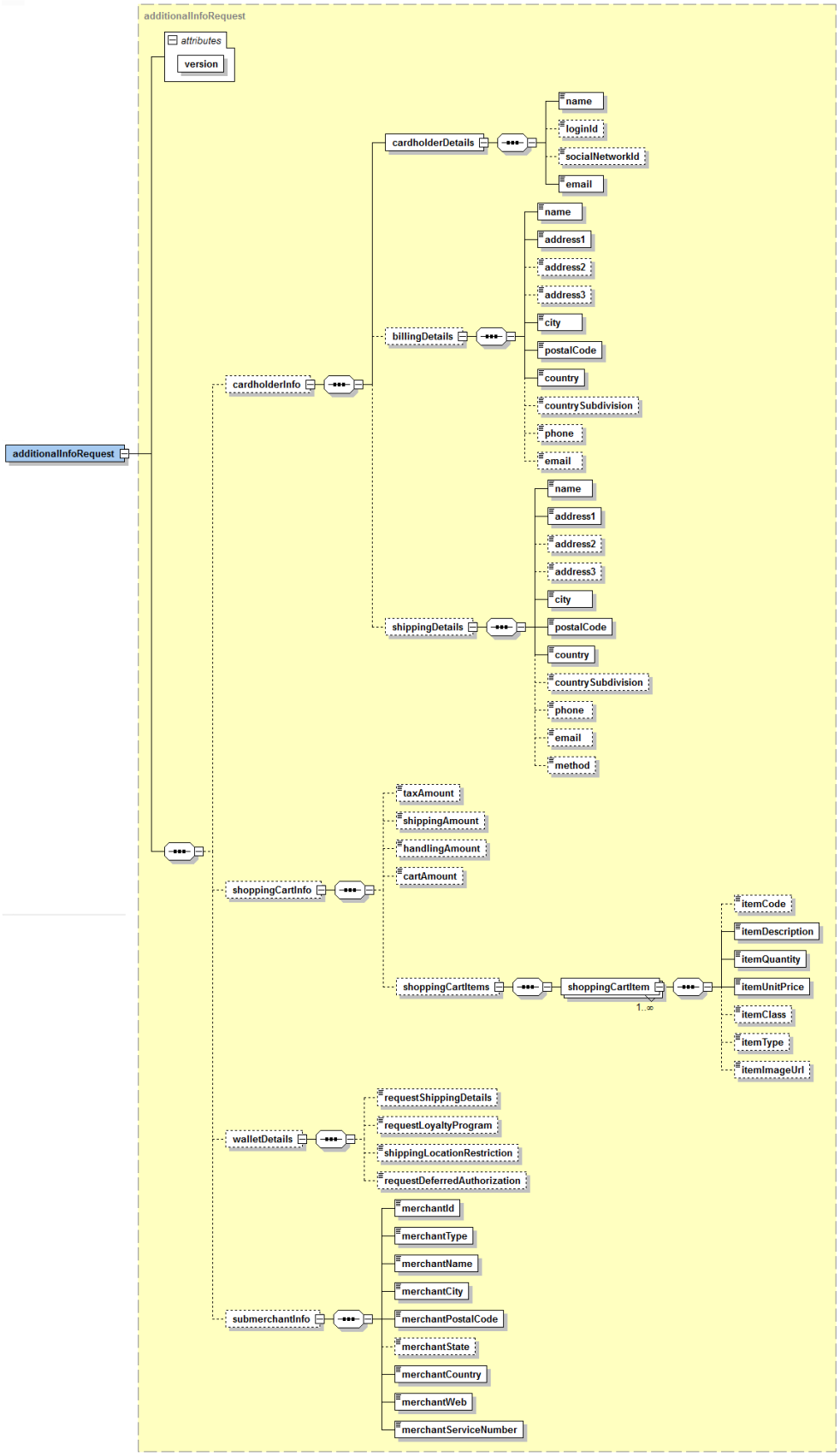
9.3 Annex no. 3 – ADDINFO field format

List of element types

Název typu	Popis
Composite type	The element is composed of more elements of various types.
Amount	The number of max. 12 digits. The value must be stated in the smallest monetary unit of a given currency without decimal point.

9.3.1 Input parameter “ADDINFO”

9.3.1.1 Elements description



Element	Description	M/O[2]	Type
additionalInfoRequest	The main element containing all requested information.	M	Composite type
version="x.x"	*A component part is an attribute containing information about version of the used template.*	*M*	*Numeric type in the format e.g. "1.0".*
Customer´s data used in the anti-fraud system			
cardHolderInfo	Customer information	O	Composite type
cardHolderDetail	Basic information about customer	M	Composite type
name	Card holder name	M	Text, max. 255 characters
loginId	LoginID into e-shopu	O	Text, max. 255 characters
socialNetworkId	LoginID into e-shop if used login via social network (Facebook, Google ...)	O	Text, max. 255 characters
email	Card holder's e-mail	M	E-mail, max. 255 characters
phone	Phone number	O	Text, max. 20 characters
clientIpAddress	Card holder's e-mail IP address	O	Text, max. 255 characters
billingDetails	Billing address	O	Composite type
name	Name	M	Text, max. 255 characters
address1	Street – 1. line	M	Text, max. 255 characters
address2	Street – 2. line	O	Text, max. 255 characters
address3	Street – 3. line	O	Text, max. 255 characters
city	City	M	Text, max. 255 characters
postalCode	Postal code / ZIP	M	Text, max. 255 characters
country	Country	M	Text, max. 255 characters
countrySubdivision	Country subdivision	O	Text, max. 255 characters
phone	Phone number	O	Text, max. 20 characters
email	E-mail	O	E-mail, max. 255 characters
shippingDetails	Shipping address	O	Composite type
name	Name	M	Text, max. 255 characters
address1	Street – 1. line	M	Text, max. 255 characters
address2	Street – 2. line	O	Text, max. 255 characters
address3	Street – 3. line	O	Text, max. 255 characters
city	City	M	Text, max. 255 characters
postalCode	Postal code / ZIP	M	Text, max. 255 characters
country	Country	M	Text, max. 255 characters
countrySubdivision	Country subdivision	O	Text, max. 255 characters
phone	Phone number	O	Text, max. 20 characters
email	E-mail	O	E-mail, max. 255 characters
method	Delivery method `personal pick-up, courier, electronic delivery …`	O	Text, max. 255 characters
Basket data used in the anti-fraud system and electronic wallets			
shoppingCartInfo	Element containing information about the basket	O	Composite type

[2] M – mandatory, O – optional

taxAmount	VAT amount	O	Amount
shippingAmount	Shipping amount	O	Amount
handlingAmount	Handling amount	O	Amount
cartAmount	VAT-exlusive basket net value. Value is calculated as: `(shoppingCartItem1[itemQuantity] * shoppingCartItem1[itemUnitPrice]) + (shoppingCartItem2[itemQuantity] * shoppingCartItem2[itemUnitPrice]) + …`	O	Amount
shoppingCartItems	Individual items in the basket. It is possible to give more items.	M	Composite type
shoppingCartItem	Basket item	M	Composite type
itemCode	Item code, e.g. "item 1"	O	Text, max. 20 characters
itemDescription	Item description	M	Text, max. 50 characters
itemQuantity	Number of items	M	Number, max. 12 digits
itemUnitPrice	VAT-exclusive unit price	M	Amount
itemClass	Item class, e.g. "class A"	O	Text, max. 20 characters
itemType	Item type, e.g. "men´s clothing"	O	Text, max. 20 characters
itemImageUrl	Complete URL path to item picture. When using MasterPass wallet, an item picture is displayed next to the item.	O	URL, max. 2000 characters
Data section when using any of electronic wallets			
walletDetails	Element adjusting possibilities of the wallet	O	Composite type
requestShippingDetails	Switch defining, if information about delivery address is demanded in the response	O	true/false
requestLoyaltyProgram	Switch defining, if information about loyalty programme is demanded in the response	O	true/false
shippingLocationRestriction	List of countries supported for delivery	O	Limitation of delivery address choice. Supported values: CZ – Czech Republic SK – Slovakia HU – Hungary EU – European Union US – USA WW – whole world (no limits) Default value is set according to the bank seat. In case of a request to deliver to other countries, please contact our application support.
requestDeferredAuthorization	Element setting to "true" enables to suspend order processing in the GP webpay system and to request finalization data from the merchant	O	true/false
requestCardsDetails	Request for sending payment card/cards detail in the response	O	true/false

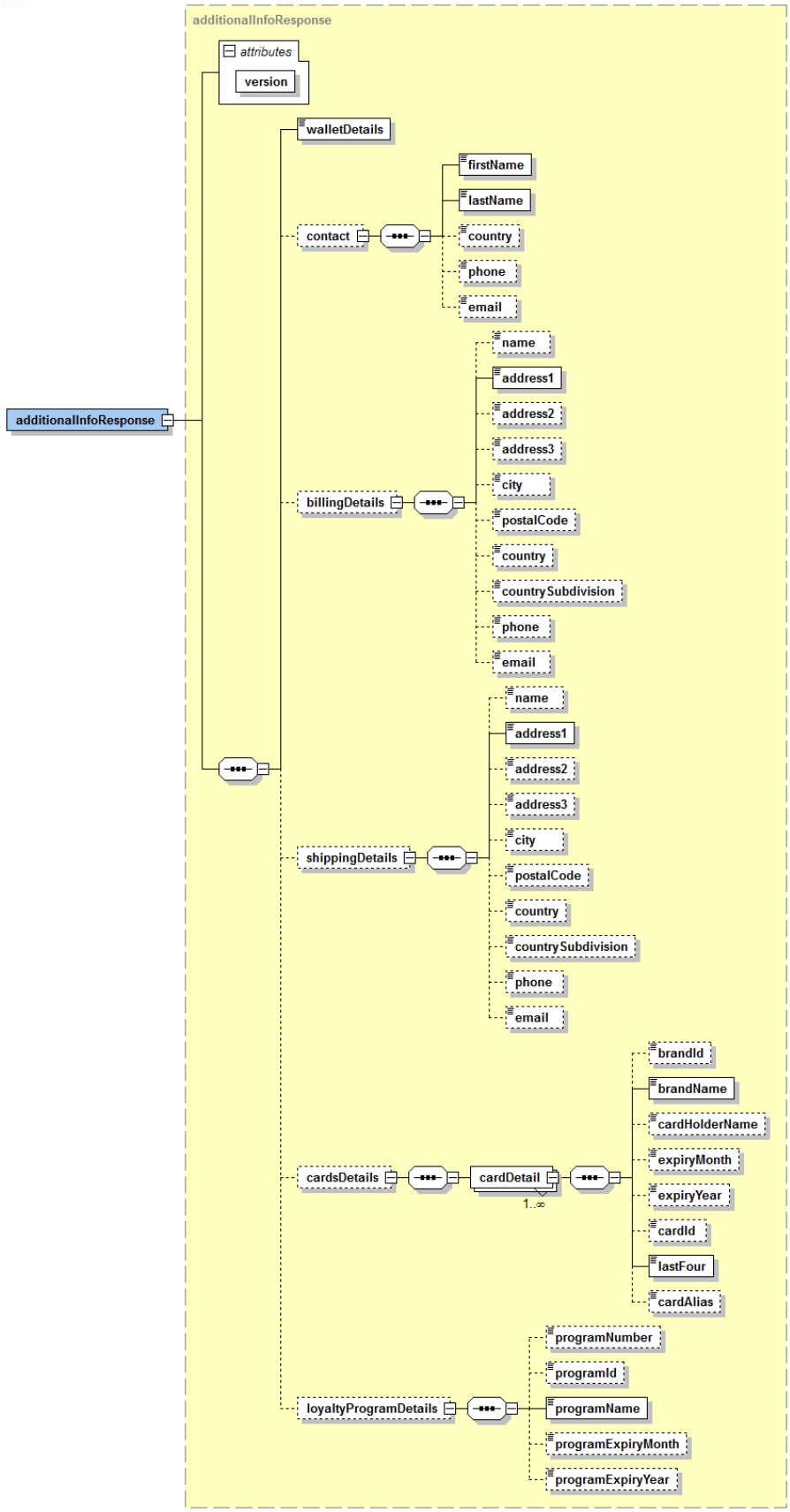
Data section for large payment services providers			
submerchantInfo	Information about merchant's realizing transactions through a payment aggregator (payment facilitator model)	O	Composite type
merchantId	A number assigned to each merchant	M	Max. 15 digits
merchantType	Merchant's MCC code	M	4 digits
merchantName	Merchant name The final name of the merchant is a composite name aggregator and merchant	M	Max. 22 characters ASCII x20-x7E
merchantStreet	Street	M	Max. 25 characters ASCII x20-x7E
merchantCity	City	M	Max. 13 characters ASCII x20-x7E
merchantPostalCode	Postal code / ZIP	M	Max. 10 characters
merchantState	State	O	Max. 3 characters
merchantCountry	Country code – ISO 3166-1 Alpha-2	M	2 characters
merchantWeb	Merchant's web page URL	M	Max. 25 characters ASCII x20-x7E
merchantServiceNumber	Merchant's phone number – customer support	M	13 digits
Request for additional information in response			
requestReturnInfo	Request for additional information in response	O	Composite type
requestCardsDetails	Request for used card information	O	true/false

9.3.1.2 Parameter scheme

GPwebpayAdditionalInfoRequest_v.3.xsd

9.3.2 Return parameter “ADDINFO”

9.3.2.1 Elements description



Element	Description	M/O	Type
additionalInfoResponse	The main element containing all requested information.	M	Composite type
version="x.x"	*A component part is an attribute containing information about version of the used template.*	*M*	*Numeric type in the format e.g. "1.0".*
Information about the used electronic wallet			
walletDetails	Information about the used wallet. Currently supported values : MPS	M	Text, max. 255 characters
Data gained from the electronic wallet			
contact	Cardholder info	O	Composite type
firstName	Name	M	Text, max. 255 characters
lastName	Surname	M	Text, max. 255 characters
country	Country	M	Text, max. 255 characters
phone	Phone	O	Text, max. 20 characters
email	E-mail	O	Text, max. 255 characters
billingDetails	Billing address	O	Composite type
name	Name	O	Text, max. 255 characters
address1	Street – 1. Line	M	Text, max. 255 characters
address2	Street – 2. Line	O	Text, max. 255 characters
address3	Street – 3. Line	O	Text, max. 255 characters
city	City	M	Text, max. 255 characters
postalCode	Postal code / ZIP	O	Text, max. 255 characters
country	Country	M	Text, max. 255 characters
countrySubdivision	Country subdivision	O	Text, max. 255 characters
phone	Phone	O	Text, max. 20 characters
email	E-mail	O	Text, max. 255 characters
shippingDetails	Shipping address	O	Composite type
name	Name	O	Text, max. 255 characters
address1	Street – 1. line	M	Text, max. 255 characters
address2	Street – 2. line	O	Text, max. 255 characters
address3	Street – 3. line	O	Text, max. 255 characters
city	City	M	Text, max. 255 characters
postalCode	Postal code / ZIP	O	Text, max. 255 characters
country	Country	M	Text, max. 255 characters
countrySubdivision	Country subdivision	O	Text, max. 255 characters
phone	Phone	O	Text, max. 20 characters
email	E-mail	O	Text, max. 255 characters
Data gained from the electronic wallet			
cardsDetails	Details about cards registered in electronic wallet and meeting conditions given in the input request.	O	Composite type
cardDetail	Card detail; there can be more of them (when using electronic wallet)	M	Composite type
brandId	Card association ID	O	Text, max. 255 characters
brandName	Name of the card association	M	Text, max. 255 characters
cardHolderName	Cardholder name	O	Text, max. 255 characters
expiryMonth	Month of card expiration	O	1-2 digits

expiryYear	Year of card expiration	O	4 digits
cardId	Card ID in the electronic wallet	O	Text, max. 255 characters
lastFour	Last 4 digits of the card number	M	4 digits
cardAlias	Card alias in the electronic wallet	O	Text, max. 255 characters
Data gained from the electronic wallet			
loyaltyProgramDetails	Information about loyalty programme	O	Composite type
programNumber	Programme number	O	Text, max. 255 characters
programId	Programme ID	O	Text, max. 255 characters
programName	Programme name	M	Text, max. 255 characters
programExpiryMonth	Month of programme termination	O	Number, 1-12
programExpiryYear	Year of programme termination	O	Number, 2014-2099

9.3.2.2 Parameter scheme

9.4 Addendum no. 1 – BASE64 encoding / decoding

Base64 is an encoding algorithm used to encode any binary data to a text form which can be easily printed and transmitted.

The result of the Base64 encoding can be transmitted without any risk of the data being converted and destroyed this way.

Base64 encoding uses the defined alphabet consisting of 65 US-ASCII characters (64 characters and space). See the following table:

Value	Encoding	Value	Encoding	Value	Encoding	Value	Encoding
0	A	17	R	34	i	51	z
1	B	18	S	35	j	52	0
2	C	19	T	36	k	53	1
3	D	20	U	37	l	54	2
4	E	21	V	38	m	55	3
5	F	22	W	39	n	56	4
6	G	23	X	40	o	57	5
7	H	24	Y	41	p	58	6
8	I	25	Z	42	q	59	7
9	J	26	a	43	r	60	8
10	K	27	b	44	s	61	9
11	L	28	c	45	t	62	+
12	M	29	d	46	u	63	/
13	N	30	e	47	v		
14	O	31	f	48	w	(pad)	=
15	P	32	g	49	x		
16	Q	33	h	50	y		

The source data are converted into the binary system as a flow of input bits (1 character equals 8 bits). The input flow is divided into groups of 6 bits and the values are converted according to the codes from the encoding table.

Every 3 input characters (3 x 8 = 24) are encoded as 4 output characters (24 / 6 = 4). If there are less then 24 bits at the end of the input data after it is divided, zero bits are appended to the input data from the right side. Zero bits appended to the input data are indicated with “**=**”.

Decoding of base64 encoded data is a process exactly reverted to base64 encoding. A flow of bits is extracted from the encoded data using the encoding table. The flow is then divided into groups of 8 bits, and the groups are converted back to the original form of the input data.

See RFC 3548 for a detailed description of base64 encoding.

9.5 Addendum no. 2 – Documentation and information sources

- ISO 639-1:2002 Codes for the representation of names of languages

  Part 1: Alpha-2 code
- ISO 639-2:1998 Codes for the representation of names of languages

  Part 2: Alpha-3 code
- ISO 4217:2001 Codes for the representation of currencies and funds
- RFC 3066 – Tags for the Identification of Languages

9.6 Addendum no. 3 – Maximum length of MERORDERNUM field

Maximum length of **MERORDERNUM** for particular banks as displayed in reports devoted for merchants:

Bank	Max. number of digits in MERORDERNUM displayed in the bank's report
Komerční banka	16
ČSOB CZ	
Raiffeisen bank	10
UniCredit bank	12
ČSOB SK	
ČSAS	