cPay – Merchant Integration Specification

CONTENTS

Using this specification

Purpose

The purpose of this document is to introduce cPay system as a payment platform to the e-commerce merchants. The document specifies the payment process using cPay as a payment platform. Also defines the interaction between the e-shop and cPay system (cPay).

This specification is in correlation with document “cPay_Testing Procedures and Production Implementation.pdf”, where are explained all steps the merchant should take during testing period and implementation in production.

Audience

The document intends for technical departments in organizations that provide e-commerce services and technical information in order to integrate e-shops to the cPay system successfully.

Introduction

Each e-commerce provider should offer to the customers reliable and secure payment procedures. The procedures should ensure fast payments processing and minimizing the possibility for complains at same time. This would result in faster delivery of goods and encourage customer loyalty.

cPay offers a reliable and easy-to-integrate platform for online card payments. Thus, the e-shop handles the shopping cart and cPay handles the payment of the selected goods. cPay offers the merchants variety of payment types that can be performed.

- One step payments, with purchase transaction
- Two step payments, with Preauthorization and Completion transactions (Appendix D)
- Installments (Appendix B)
- Recurring Transactions (Appendix C)
- Credit transactions
- Transactions with registered cards (Appendix E)

The e-shop securely redirects the customer to cPay sending in parallel the payment parameters.

The customer provides card details only at cPay.

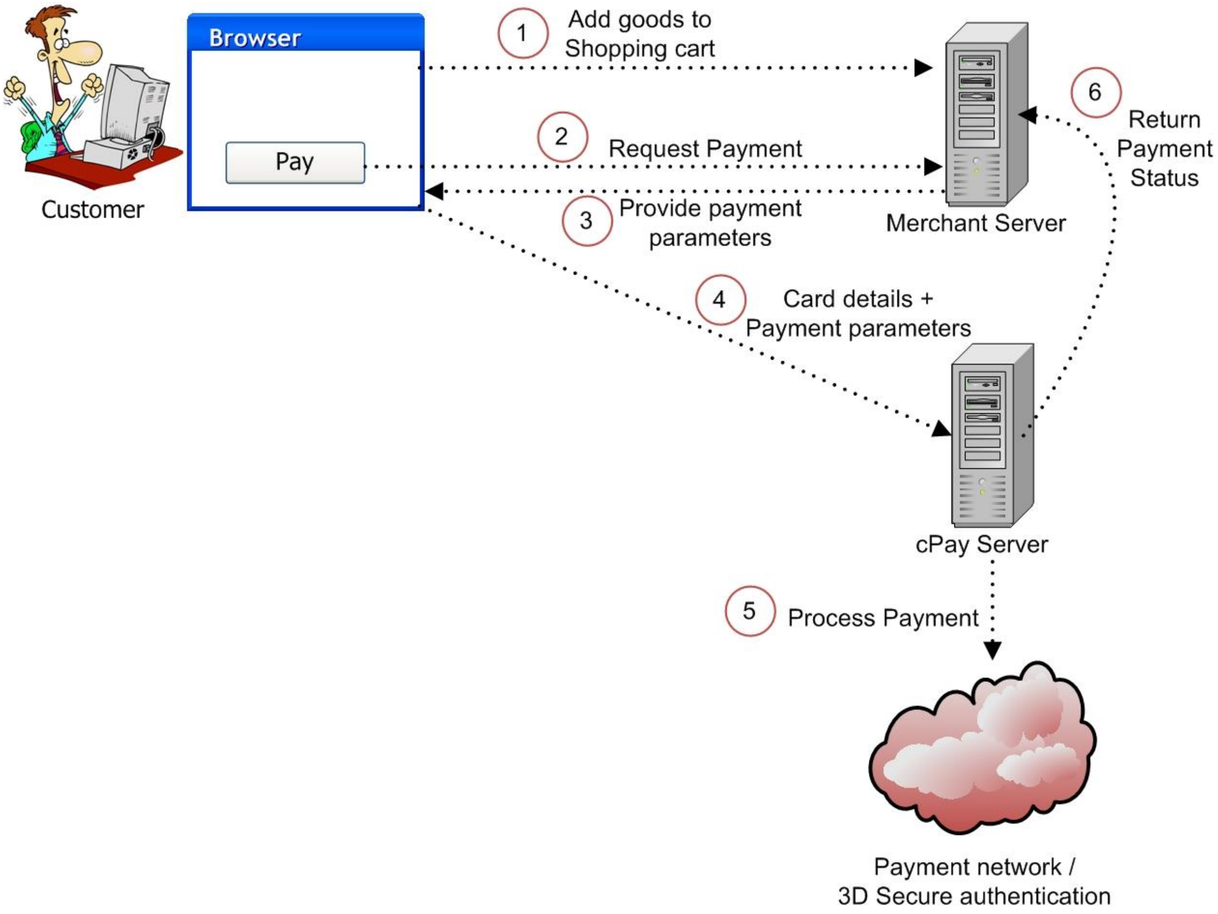
Supported card brands are MasterCard, Maestro, Visa, Diners club, Domestic cards. It is recommended the merchant on their site to provide information which card brands are accepted.

cPay supports the traditional cardholder authentication with CVV2/CVC2 as well as 3D Secure authentication. This ensures additional security in processing of card operations and decreases fraud payments.

- Authentication of CVV2/CVC2 (3-digit number at the back of the card) is performed by the issuer bank after submitting the authorization request from cPay.

- 3D (Visa/MasterCard) Secure Authentication is performed if payment card participates in 3D secure program. In this case, the authentication is performed by the issuer bank through Visa/MasterCard network.. For 3D authentication, the cardholder should provide additional password to the issuer bank through Visa/MasterCard network. Depending of the result of this authentication, cPay will or will not proceed with the authorization request i.e. payment transaction.

Payment Process



Step	Description
Step 1.	The customer selects goods at the e-shop and adds them to the shopping cart
Step 2.	The customer clicks on the “Pay” button.
Step 3.	The e-shop redirects the payment parameters to cPay via the customer’s browser. The customer now interacts with cPay.
Step 4.	The customer enters card information at cPay.
Step 5.	Additional cardholder authentication is performed If the customer’s card is registered for 3D Secure cPay processes the card payment.
Step 6.	cPay redirects the customer’s browser back to the e-shop to a success page or failure page according to the payment status.

Payment Parameters

A payment URL is provided to the merchant in order to redirect the customer's browser:

Parameter	Description
PaymentURL	URL that opens simple payment card entry form with mandatory fields to fulfulled (respectively for Macedonian and English language): https://www.cpay.com.mk/client/Page/default.aspx?xml_id=/mk-MK/.loginToPay/.simple/ https://www.cpay.com.mk/client/Page/default.aspx?xml_id=/en-US/.loginToPay/.simple/

The e-shop should provide several parameters to cPay for each payment (using UTF-8 encoding). Please consider the Format/Length of all parameters in order payments to be correctly and successfully processed.

Parameter	Format	Length	Required	Description
AmountToPay	numeric 0-9	12	Yes	Amount of the payment, multiplied by 100. e.g., 1 MKD should be represented with the value 100. **Note: Last two digits must always be 00 e.g. xxxx00. Payment on zero Amount is not allowed.**
AmountCurrency	ISO 4217 Code MKD	3	Yes	Currency of the payment. Payments in Macedonia may be conducted only in Denars, so this parameter always will be MKD.
Details1	alphanumeric 0-9, a-z, A-Z,SC*	32	Yes	Details/Description about the payment.
Details2	alphanumeric 0-9, a-z, A-Z	10	Yes	More details about the payment. The unique payment reference at the e-shop should be passed as Details2. **Details2 should have unique value for each payment.** The merchants should use this parameter to match the cPay responses with the payments in their system.
PayToMerchant	numeric 0-9	100	Yes	Merchant's id at cPay. The value of this parameter is included in the template redirect code that merchant download from the merchant module of cPay Payment Portal. **The value of this parameter must not be changed.**
MerchantName	alphanumeric 0-9, a-z, A-Z,SC*	200	Yes	Merchant's name. The value of this parameter is included in the template redirect code that merchant download from the merchant module of cPay Payment Portal. **The value of this parameter must not be changed.**
PaymentOKURL	alphanumeric 0-9, a-z, A-Z, SC*	500	Yes	After the payment is processed successfully, cPay redirects the cardholder's browser back to the e-shop at the URL PaymentOKURL. The push notifications for successful payments are also sent on PaymentOKURL The value for PaymentOKURL should be a valid value. In the downloaded, redirect template code from the merchant module on cPay, the value for PaymentOKURL: http://merchantOKurl.com is just an example.

PaymentFailURL	alphanumeric 0-9, a-z, A-Z, SC*	500	Yes	After the payment is processed with failure, cPay redirects the cardholder's browser back to the e-shop at the URL PaymentFailURL. The push notifications for unsuccessful payments are also sent on PaymentFailURL The value for PaymentFailURL should be a valid value. In the downloaded redirect template code from the merchant module on cPay, the value for PaymentFailURL: http://merchantFailurl.com is just an example.
OriginalAmount	numeric 0-9	12	No	The amount as displayed at the merchant's web site, represented in specific (original) currency. This field is displayed on cPay for informative purposes. However, the payment is performed in the currency of the acquiring bank (MKD) according to the parameters AmountToPay and Amount Currency. This parameter is informative - it should NOT be multiplied by 100. It is displayed exactly as provided by the merchant. Currency conversion is responsibility of the merchant.
OriginalCurrency	ISO 4217 Code e.g EUR	3	No	The currency, corresponding to the parameter OriginalAmount. This field is displayed on cPay for informative purposes.

Fee	alphanumeric
	0-9, a-z, A-Z, %
		16	No	Used to provide information to the buyer for the fee added to the original amount by the merchant.
				Example:
				Fee="10%"
				Fee="10den"

CRef	alphanumeric 0-9, a-z, A-Z	50	No	The field "CRef" is used for card registration for further re-usage. Please refer to Appendix E for more details about transaction with registered cards. **Note: Requires additional contract/agreement with bank.**
TransactionType	alphanumeric 0-9, a-z, A-Z	3	No	This field is used to perform other transaction types different from the standard purchase transaction. To perform credit (Refund) transaction TransactionType=004 should be sent. With this type of transaction, the cardholder account is credited with the amount sent in the request. **Note: Requires additional contract/agreement with bank.**
Installment	numeric 02-99	2	No	The Installment field carries the information for the number of installments (02-99) in the Installment transaction. Please refer to Appendix B for more details. **Note: Requires additional contract/agreement with bank.**
RPRef	alphanumeric 0-9, a-z, A-Z	50	No	The RPRef field carries the information for the Recurring Payment service. Please refer to Appendix C for more details for Recurring Payment transactions. **Note: Requires additional contract/agreement with bank.**
FirstName	alphanumeric 0-9, a-z, A-Z	64	No	Cardholder's first name
LastName	alphanumeric 0-9, a-z, A-Z	64	No	Cardholder's last name

Address	alphanumeric 0-9, a-z, A-Z, SC*	50	No**	Cardholder’s billing address
City	alphanumeric 0-9, a-z, A-Z	50	No	Cardholder’s billing address - city
Zip	numeric 0-9	16	No	Cardholder’s billing address - zip or postal code
Country	numeric 0-9, ISO 3166-1	3	No	Cardholder’s billing address – country according ISO 3166-1 numeric three-digit country code, e.g. 807 for Macedonia
Telephone	numeric 0-9	15	No**	Cardholder’s mobile phone containing **Country code** and *subscriber* sections of the number, e.g. **389**_7yxxxxxx_, where x, y-numeric value. Please refer to ITU-E.164
Email	alphanumeric 0-9, a-z, A-Z, SC*	256	No**	Cardholder’s e-mail, shall meet requirement of IETF RFC 5322, e.g. test@gmail.com
CheckSumHeader	alphanumeric 0-9, a-z, A-Z	1024	YES	Header for the CheckSum. Explained in the Appendix A. **Note: Parameters with empty values should not be added in the checksum header**
CheckSum	alphanumeric 0-9, a-f, A-F	32	YES	Md5 hash value, generated by the concatenation of those parameters that are included in the redirect code and the merchant checksum authentication key. The checksum algorithm is specified in details in Appendix A The checksum algorithm builds an MD5 hash value over the concatenation of the following parameters: ▪ AmountToPay ▪ PayToMerchant ▪ MerchantName ▪ AmountCurrency ▪ Details1 ▪ Details2 ▪ PaymentFailURL ▪ PaymentOKURL ▪ Fee (if present) ▪ CRef (if present) ▪ TransactionType (if present) ▪ Installment (if present) ▪ RPRef (if present) ▪ OriginalAmount (if present) ▪ OriginalCurrency (if present) ▪ FirstName (if present) ▪ LastName (if present) ▪ Address (if present) ▪ City (if present) ▪ Zip (if present) ▪ Country (if present) ▪ Telephone (if present) ▪ Email (if present) ▪ merchant’s checksum authentication key The order of the parameters is specified in the CheckSumHeader. The checksum authentication key: ✓ is assigned to the merchant ✓ is kept at both sides – cPay and merchant ✓ is never sent in a message

			✓ is used to generate MD5 value ✓ in the test period the default merchant's checksum authentication key is TEST_PASS The MD5 value is generated based on the specifics: ✓ The initial text is encoded in UTF-8 ✓ The hash value (16 bytes) is represented in hex as 32 characters (0-9 and a-f, A-F)

*SC = Space character, @, -, /, ?, =, &.

Note: Two consecutive "@" characters are not allowed.

**These parameters are not required to be present in the redirect code, but are recommended in order to increase frictionless 3DS2 transactions, avoid false declines, and to increase the authorization approval rates. Also other billing information (city, zip, country) can be included in the redirect code.

cPay generates a response and returns it to PaymentOKURL or PaymentFailURL depending on the payment result. The response is sent to the merchant's server trough two channels:

- ✓ HTTP request directly from cPay to PaymentOKURL or PaymentFailURL (PUSH messages)
- ✓ Redirect of customer's browser to PaymentOKURL or PaymentFailURL (Browser redirect)

The response echoes the parameters from the request and contains additional parameters:

Parameter	Format	Length	Description
cPayPaymentRef	numeric 0-9	10	Reference number of the card payment in cPay. cPay generates this number only when the customer submits the card details on the first screen on cPay payment process; otherwise it is not present.
ReturnCheckSumHeader	alphanumeric 0-9, a-z, A-Z, SC*	1024	Header for the ReturnCheckSum. Explained in the Appendix A. The places of the first and the second parameter from the merchant's checksum header are switched in the return checksum header. Thus, the ReturnChecksum will always differ from the initial checksum, provided by the merchant.
ReturnCheckSum	alphanumeric 0-9, a-f, A-F	32	MD5 hash value, generated by the concatenation of all present parameters and a merchant checksum authentication key. The checksum algorithm is specified in details in Appendix A The checksum algorithm builds an MD5 hash value over the concatenation of the following: ▪ PayToMerchant ▪ AmountToPay ▪ MerchantName ▪ AmountCurrency ▪ Details1 ▪ Details2 ▪ PaymentFailURL ▪ PaymentOKURL ▪ Fee (if present) ▪ CRef (if present) ▪ TransactionType (if present)

	▪ Installment (if present) ▪ RPRef (if present) ▪ OriginalAmount (if present) ▪ OriginalCurrency (if present) ▪ FirstName (if present) ▪ LastName (if present) ▪ Address (if present) ▪ City (if present) ▪ Zip (if present) ▪ Country (if present) ▪ Telephone (if present) ▪ Email (if present) ▪ cPayPaymentRef (if present) ▪ merchant's checksum authentication key The order of the parameters is specified in the ReturnCheckSumHeader. The checksum authentication key is the same as the one, used for the Checksum in the request. If the user cancels the payment at the first screen of cPay, no payment reference is generated for that payment. In this case the parameter cPayPaymentRef is neither sent to the PaymentFailURL, nor included in the ReturnChecksum.

*SC = Space character, @,/,-,?,=,&.

In response, cPay will echo all request parameters, together with additional return parameters including a "ReturnCheckSum". The merchant must validate the ReturnCheckSum in order to make sure the response corresponds to the original transaction started on the website of the merchant. The merchant should use the field Details2, AmountToPay, to uniquely identify transactions started at his website. The merchant also must validate the format and the parameters return in cPay response*.

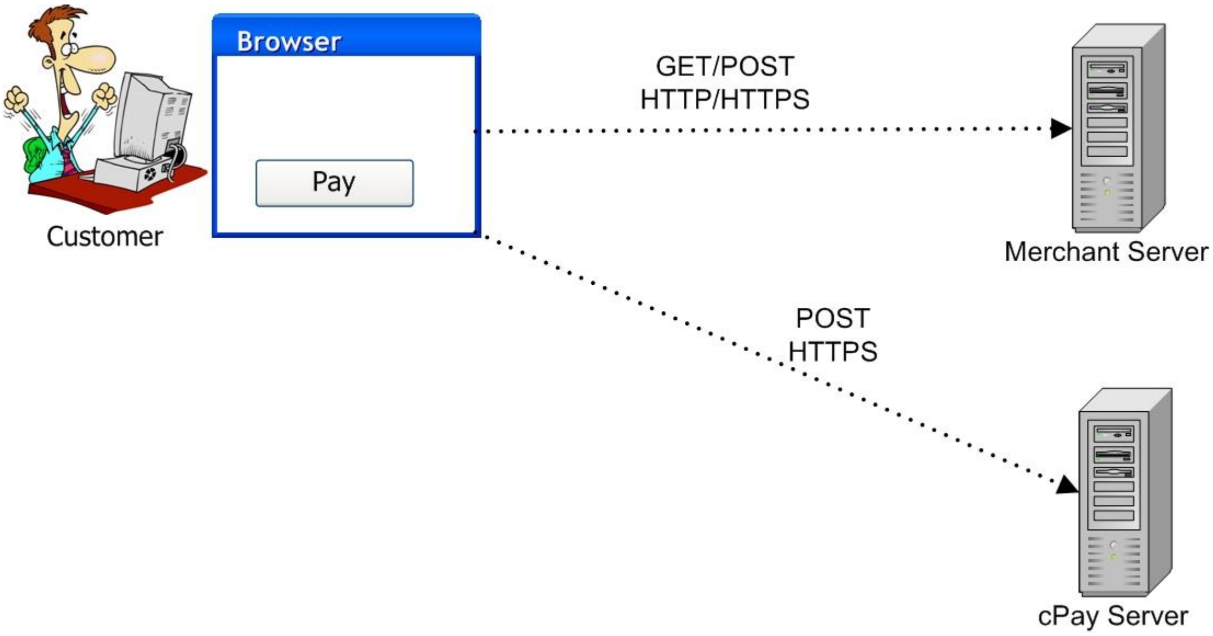
The merchant may also consider adding their own, unique parameter in the payment OK/FAIL URLs. For example, this may be the merchant's own transaction identifier, which can be pass as a GET parameter in the OK/FAIL URL, so when a buyer is directed back to the store, the buyer's browser will carry the merchant's own identifier, as an additional verification.

The above notes must be considered to eliminate possible fraud transactions that might occur with direct access to PaymentOKURL (there will be no requests to cPay) and the merchant e-shop might allow the payment, and the goods/services might be delivered.

* cPay reserves the right to return additional parameters in response, in accordance with this specification (if present, these additional parameters will be included in ReturnCheckSumHeader, InputString, and in calculation for ReturnCheckSum).

Technical Details

Communication Protocols



The communication between the customer and the e-shop varies according to the e-shop's architecture.

The communication between the customer and cPay is always encrypted in HTTPS with server certificate only.

The HTTP parameters must be sent to cPay in UTF-8 encoding using the method POST.

Client browsers that are redirected to the cPay payment portal MUST support the latest security protocols i.e. TLS 1.2 with appropriate AES 256 cyphers in order cPay payment page to be loaded.

Also, if the merchant's website is HTTPS, then it is necessary to support the latest security protocols, i.e. TLS 1.2 with appropriate AES 256 cyphers, so push messages to be successfully delivered to merchant server.

Obsolete and not secure protocols TLS 1.0, TLS 1.1 do not support cPay.

Note: The technical implementation of the web site is subject to changes dictated by the security standards PCI DSS, VISA, MasterCard, and according to those standards, the interface to the cPay gateway needs to be upgraded when necessary.

Passing Parameters

When the customer presses the “Pay” button, the POST parameters are sent to cPay.

An easy way to implement this functionality is to insert fields of type hidden at the e-shop payment form. These fields would contain values about the merchant, amount, payment details – all required and some optional parameters listed in the Payment Parameters section.

For example:

```
<form action="https://www.cpay.com.mk/client/Page/default.aspx?xml_id=/mk-MK/.loginToPay/" method="post" ID="Form1">
.......
<input type="hidden" id="AmountToPay" name="AmountToPay" value="5000"/>
<input type="hidden" id="AmountCurrency" name="AmountCurrency" value="MKD"/>
<input type="hidden" id="Details1" name="Details1" value="Invoice"/><!-- purchase info -->
<input type="hidden" id="Details2" name="Details2" value="99"/><!-- payment reference at e-shop -->
<input type="hidden" id="PayToMerchant" name="PayToMerchant" value="123745"/>
<input type="hidden" id="MerchantName" name="MerchantName" value=”MerchantSample"/>
<input type="hidden" id="PaymentOKURL" name="PaymentOKURL" value="http://sampleeshop/e-shopOK.html"/>
<input type="hidden" id="PaymentFailURL" name="PaymentFailURL" value="http://sampleeshop/e-shopCancel.html"/>
.......
</form>
```

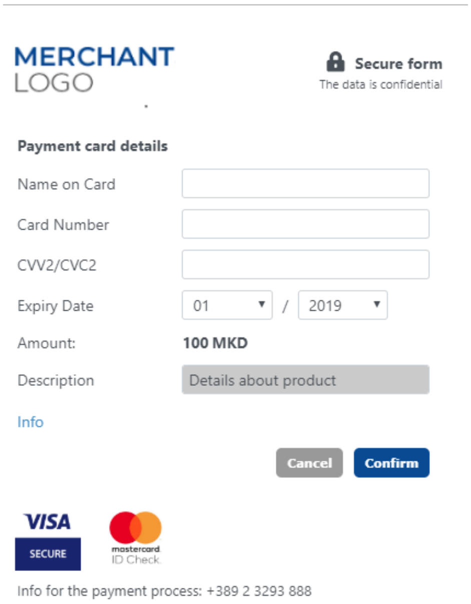
Note: Do not edit/remove any parameters in HTTP request header while performing redirect to cPay.

The parameter **Referer** MUST be present in HTTP Request Headers with value of the merchant URL from where the client is redirected to cPay payment portal. This is not a parameter from the redirect code, but is generated from the browser. Commonly this parameter is sent in every redirect, but if the client or the merchant has disable sending of HTTP **Referer**, it will be empty in the request and cPay cannot validate the redirect, resulting with unsuccessful payment.

cPay Card entry page

When performing a transaction, the buyer is redirected to cPay payment website, which is highly secure and implements industry-leading technology to keep the cardholder information safe.

The following are the first and second default screens of payment that are presented to the client:



MERCHANT LOGO

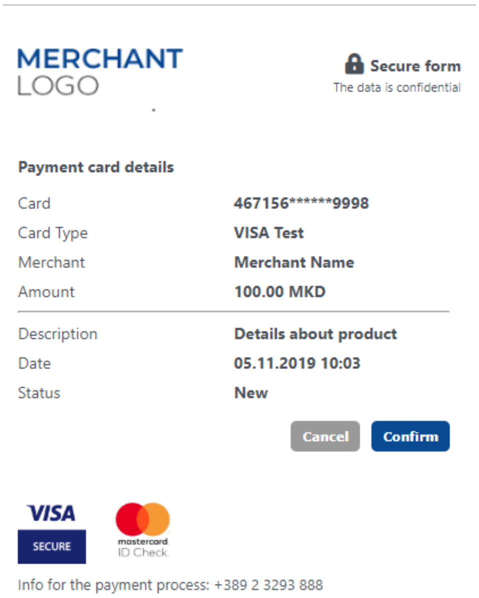
Secure form
The data is confidential

Payment card details

Name on Card	
Card Number	
CVV2/CVC2	
Expiry Date	01 / 2019
Amount:	100 MKD
Description	Details about product

Info

Info for the payment process: +389 2 3293 888



MERCHANT LOGO

Secure form
The data is confidential

Payment card details

Card	467156******9998
Card Type	VISA Test
Merchant	Merchant Name
Amount	100.00 MKD
Description	Details about product
Date	05.11.2019 10:03
Status	New

Info for the payment process: +389 2 3293 888

There is option merchant to customize the default payment screens, so the client will not "feel" that is leaving the merchant's page in the payment process. The merchant can make his own design solution and retain the style of his e-shop simply by sending a modified .CSS file. For that purpose please find the CSS code archive for the payment form in the “Redirect Code” menu in the merchant module, which contains the code for the payment form style. The html files for first and second payment screen are present in the archive file, and the modification can be done only with merchant-styles.css file.

From the existing payment form, it is necessary to keep the data entry fields and the options for "Cancel" and "Confirm", as well as "Secure form" notification and Visa Secure / MasterCard ID Check logos. These logos are present in the CSS archive and you can set them also on your web page.

After customization is done, only the merchant-styles.css file should be sent to the bank representative and the bank should send modification form to CaSys along with the .CSS file.

The merchant can also send merchant logo to bank representative if it should be displayed on the payment card entry form in .jpg/.png format and the logo size can be for example 210x90px, and for square logo for example 220x220px.

***NOTE The card entry form must not be used in IFRAME window or in popup window, if used, the merchant will not be compatible with the cPay payment system and the payment form will not open on most browsers.**

****NOTE If you want to make changes to the form and text, based on the language selected, in the merchant-styles.css, you need to use identifier that notifies what language is selected.**

For example, if Macedonian version is selected and the sent URL request is like id=/mk-MK, the identifier would be class="mk", or for English language the identifier would be class="en".

Notifications about Status of Transactions

After transaction is completed on the cPay Payment System, three (3) types of notifications for the status of the transaction are submitted to the merchant in three different ways. It is necessary, notifications for the status of transactions sent by cPay payment system, to be delivered on merchant e-mail address (specified in the definition form) and to the merchant server.

The three (3) types of notifications for the status of the transaction that are sent to the merchant are explained bellow:

1. **Via buyer's browser** – according to the transaction status, the buyer will be redirected to PaymentOKURL (if it is successful payment) or PaymentFailURL (if payment is rejected). The merchant should inform the buyer about the status of the payment (whether the product/service will be delivered to the customer). This notification is the html redirect code, and contains all the parameters that the merchant has sent to cPay Payment System and additional parameters ReturnCheckSum, ReturnCheckSumHeader, and parameter cPayPaymentRef (if client submits card data).

2. **E-mail message with attached xml file and subject "order status"** – this notification cPay Payment System sends to the e-mail address specified in the definition form sent by the bank.

Example of xml file for successful transactions:

```
<?xml version="1.0"?>
-<Order xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            -<OrderStatus>
                    <StatusCode>0</StatusCode>
                    <StatusReason>Success</StatusReason>
            </OrderStatus>
            -<OrderDetails>
                    <Description>Details1</Description>
                    <Amount>AmountToPay</Amount>
                    <Currency>AmountCurrency</Currency>
            </OrderDetails>
            <OrderID>Details2</OrderID>
            <cPayPaymentRef>cPayPaymentRef</cPayPaymentRef>
</Order>
```

Example of xml file for unsuccessful transactions:

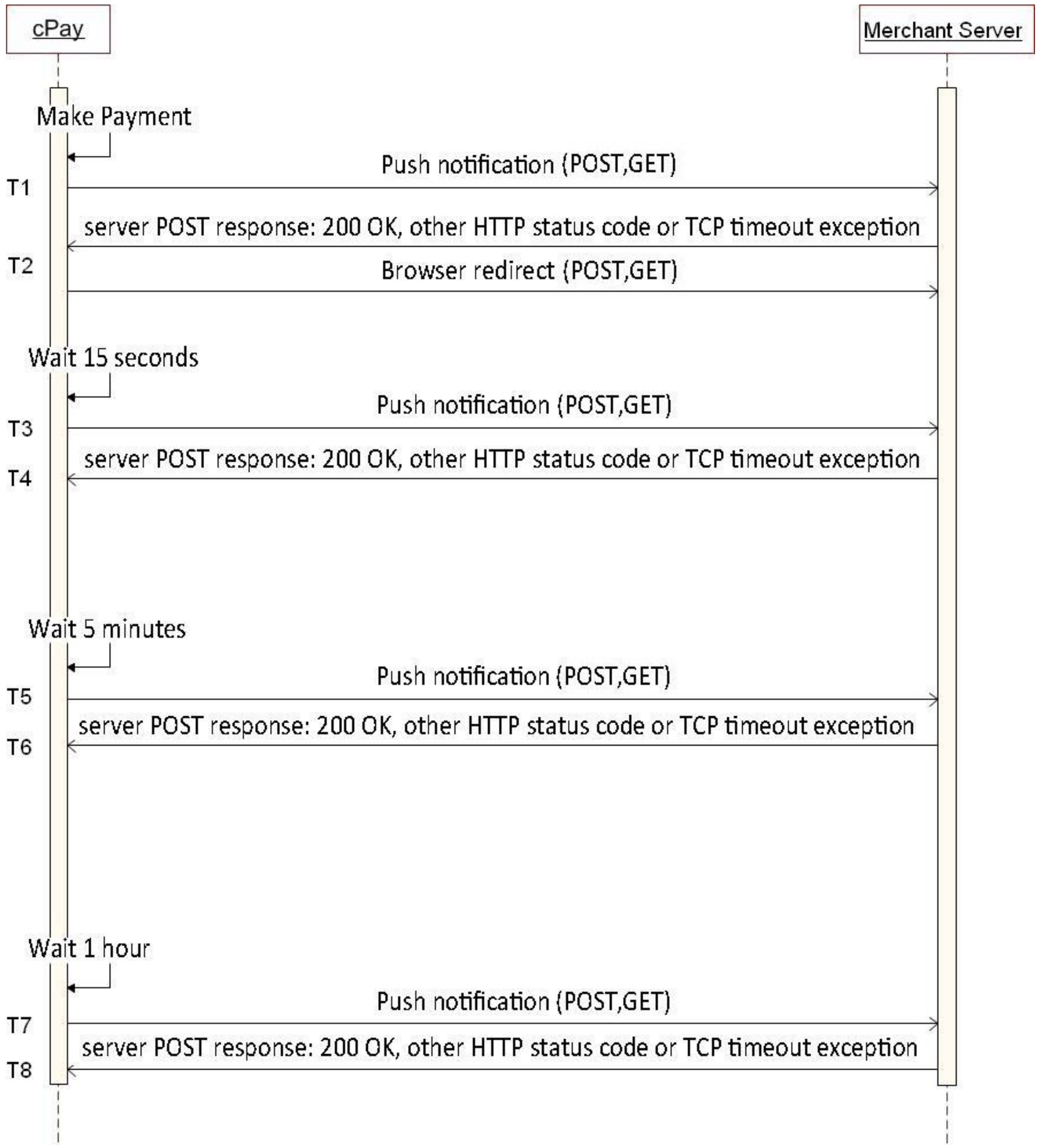
```
<?xml version="1.0"?>
-<Order xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            -<OrderStatus>
                    <StatusCode>1</StatusCode>
                    <StatusReason>Not Sent</StatusReason>
            </OrderStatus>
            -<OrderDetails>
                    <Description>Details1</Description>
                    <Amount>AmountToPay</Amount>
                    <Currency>AmountCurrency</Currency>
            </OrderDetails>
            <OrderID>Details2</OrderID>
            <cPayPaymentRef>cPayPaymentRef</cPayPaymentRef>
</Order>
```

3. **HTTP request or push messages** – this is the same HTML code, which redirects the buyer from cPay Payment system to merchant site after the payment. This request is sent directly from cPay server to merchant server (without using the buyer's browser), to PaymentOKURL or PaymentFailURL, depending of the transaction status. To receive this push messages, IP address of the merchant site (specified in the definition form sent by the Bank) should be static and should correspond to merchant's URL address.
For every change of the IP or domain name, the merchant should inform the bank in time, and the bank should send modification form to CaSys' support team accordingly. After replying to the Bank that Domain name/IP address was change in the cPay system, it is strongly recommended merchant to make one test transaction, as soon as possible, and to

check whether all three types of notifications are successfully sent to the merchant. After performing the test transaction, the merchant must inform the Bank and ecommerce team about the result of the test (successfully/unsuccessfully completed).

The dynamics of the PUSH message (HTTP request) flow is explained bellow:

- cPay send the first push message (T1) and waits for an response (the operation is synchronous). The response from the merchant server should be with HTTP status code 200 OK, but also an HTTP error code (404, 500, etc…) or an exception may occur (i.e. network error, timeout). After merchant server responds (or times out) on first PUSH message sent, cPay performs browser redirect to merchant site (T2)
- Regardless of the server response on the first PUSH message, 15 seconds after the first PUSH message sent, cPay sends the second PUSH message (T3=T1+15sec);
- If merchant server response of the second message is not with HTTP status code 200 OK, or there is no response (T4), cPay sends the third PUSH message after 5 minutes (T5=T4+5 min)
- If merchant server response of the third message is not with HTTP status code 200 OK, or there is no response (T6), cPay sends the fourth PUSH message after 1h (T7=T6+1h)
- If merchant server response of the fourth PUSH message is not with HTTP status code 200 OK, or there is no response (T8), cPay sends an e-mail message (with subject “failed merchant notification”) which contains the html code. This message is sent to the e-mail address specified in the definition form sent from the Bank.


cPay
Merchant Server
Make Payment
Push notification (POST,GET)
T1
server POST response: 200 OK, other HTTP status code or TCP timeout exception
T2
Browser redirect (POST,GET)
Wait 15 seconds
Push notification (POST,GET)
T3
server POST response: 200 OK, other HTTP status code or TCP timeout exception
T4
Wait 5 minutes
Push notification (POST,GET)
T5
server POST response: 200 OK, other HTTP status code or TCP timeout exception
T6
Wait 1 hour
Push notification (POST,GET)
T7
server POST response: 200 OK, other HTTP status code or TCP timeout exception
T8

When cPay PUSH parameters service sends information to the merchant server, it uses the POST request method of the HTTP protocol.

The PUSH service, which is the client, sends a standard header “Expect: 100-Continue”, and expects the server to either respond with an HTTP status code of 100, or send some other appropriate instruction (like redirect to a different location).

In general, the merchant should perform the following steps for each push notification or browser redirect:

- Read the POST parameters in the push notification (Read the GET/POST parameters via browser redirect to OK URL or FAIL URL).
- Check the ReturnCheckSum.
- Identify the order at the merchant system using the unique Details2, AmountToPay for that order, and mark it as successful or not.
- Return to cPay HTTP status 200 OK for every PUSH notification.
- When browser redirect from cPay back to the merchant OK/FAIL URL is performed, the merchant should display to the user a message that the order is successfully/not successfully paid, depending of the successful/unsuccessful validation of the parameters returned by cPay in the browser redirect response: ReturnCheckSum, unique Details2, AmountToPay.

Please note that the notification service handles also scenarios when the customer closes the browser very fast, before the redirect to the merchant site (the browser is closed prior to cPay receiving the transactions statuses) or the internet connection is down and in both cases the normal transaction flow is interrupted before the regular e-mail and post notifications were generated. Such cases happen rarely and for these cases, after 15 minutes from the transaction, cPay sends email (with attached xml file) and first push notification to the merchant, which is the optimal defined time. Consecutive push notifications are sent according previously explained push dynamic flow.

IMPORTANT NOTE: When merchant is defined in cPay system, merchant/developer have a period of 1 month to test cPay system in local environment, meaning the payment can be submitted to cPay from development URL and development URL can also be used in parameters **PaymentOKURL** and **PaymentFailURL.** After 1 month from merchant definition in cPay system (or after changing to production checksum authentication key – going to production mode), the payments MUST be submitted to cPay only from the domain name that is registered for e-shop. If the payment request is received from any other domain name, the payment will be declined. Also, the parameters **PaymentOKURL** and **PaymentFailURL** that are sent in the redirect code MUST correlate with the domain name that is registered for e-shop, otherwise the payment will be decline. After 1 month from merchant definition in cPay, domain restriction will apply and if merchant want to test from other domain, Bank representative must submit modification form for changing the URL. Domain name restriction rule is implemented because each virtual ecommerce terminal is defined in the system with unique Domain name, IP address and appropriate Merchant Category code (MCC) according to the products/services that are sold on the merchant web site.

Please take in consideration and following notes during implementation:

Note 1: Do not use encoded reference for ampersand sign (&amp;) in the request parameters, it will result with “system error”. CPay does not expect xml encoded character reference and does not process them. Instead of **&amp;** please use **&**.

Note 2: Do not use Cyrillic domain name in OK/FAIL URL parameters, but use the actual domain encoded as ASCII string (via Punycode transcription). For example, instead of www.пример.com send www.xn--e1afmkfd.com in the merchant OK/FAIL URLs parameters.

Note 3: The push notifications can be sent only on default http/https ports (80/443). Do not use ports other from defaults in OK/FAIL URLs parameters.

Note 4: Do not use any private IP addresses in OK/FAIL URLs parameters. Use only valid domain name in OK/FAIL URLs parameters, domain name which is registered for e-shop.

Note 5: If HTTPS encryption is used on the merchant web site, please use only valid commercial SSL certificates. Do not use self-signed certificates.

Note 6: Do not use combination of characters that are used in SQL injection and cross-site scripting. Otherwise, the IP address of the client which performs the payment will be blocked, and will not have access to cPay portal for certain period (for example do not use " ' " or "@@"). For security reasons of our system list of all restrictions cannot be provided but the following web site with examples can be used:

http://www.symantec.com/connect/articles/detection-sql-injection-and-cross-site-scripting-attacks

Payments Status

Explanation follows of the transactions status that can appear on cPay merchant module:

- **Sent / Успешна:** successful transaction, successfully processed transaction.
- **Rejected / Одбиена**: unsuccessful transaction, rejected by the card issuer or cPay
- **Authorized**: Successful pre-authorization transaction, the amount is reserved from cardholder account.
- **Completed authorization:** successful completion of previous pre-authorization transaction. Completion amount is debited from cardholder account.
- **New / Нарачка во процес** : pending transaction, transaction that is not yet confirmed by the client, it is temporary status Later, after appropriate action by the client, this status will be change to one of the final statuses (if no action is taken this status will be change to Expired).
- **Canceled / Откажана**: unsuccessful transaction, transaction that is canceled by the client
- **Blocked:** unsuccessful transaction, blocked because of set black/white list for the merchant
- **Blocked by Card Limits:** unsuccessful transaction, blocked because of set limit for amount/number of transactions in defined period.
- **System Reversal:** unsuccessful transaction, the transaction is not successfully processed in authorization and cPay system generates reversal.
- **In 3D Secure authentication / Во 3D Secure автентикација**: pending transaction, the client is redirected to the bank ACS site for verification of the 3D Secure code/password, it is temporary status. Later, after appropriate action by the client, this status will be change to one of the final statuses. If the client do not take any action or closes the browser without entering the 3D Secure code, this status will be change to Expired.
- **3D Secure authenticated / 3D Secure автентицирана:** pending transaction, the client has successful entered the 3D Secure code and has passed 3D authentication, it is temporary status. Later, this status will be change to one of the final statuses. If the client closes the browser after entering the 3D Secure code, this status is changed to Expired.
- **3D Secure authentication failed / 3D Secure автентикацијата е неуспешна**: unsuccessful transaction, the client has entered an incorrect 3D Secure code and has failed 3D Secure authentication.
- **Refunded / Рефундирана:** successful refund transaction, perform by the bank on a request from the merchant, The Refund option can also be enabled on merchant module, please contact the bank.
- **Partially Refunded / Делумно Рефундирана:** successful partially refunded transaction, performed by the bank on a request from the merchant, The partial refund option can also be enabled on merchant module, please contact the bank
- **Expired**: unsuccessful transaction. All transactions that are not completed on cPay will have this status, as explain above (previous statuses: New, In 3D authentication, 3D secure authenticated)

Appendix A –CheckSum Specification

The CheckSum in the merchant request and the ReturnChecksum is built using a secure algorithm, which includes all parameters included in the redirect code and a header, specifying the number of parameters, order of the parameters and their lengths.

Important! All request parameters should be included when building the checksum. If some request parameters are missing in the header - the payment is treated as a fraudulent transaction

The Input String

The checksum is constructed by applying the MD5 hash algorithm to an input string.

The input string consists of three parts: a header, a concatenation of parameter values and the merchant's checksum authentication key (used for calculating MD5 hash value – checksum):

Input string:

Header	**Value of Param 1**	**Value of Param 2**	**…**	**Value of Param N**	**Checksum auth. key**

For example:

08PaymentOKURL,PaymentFailURL,AmountToPay,AmountCurrency,PayToMerchant,Details1,Details2,MerchantName,025027005003010017011009https://bookstore/ok.htmlhttps://bookstore/fail.html12300MKD1000000003purchase of booksOrder 25467BookstoreTEST_PASS

And the corresponding md5 hash value (checksum) is:

34F2872495067872C7D11C4D0F6A3DE2

The Header

The header starts with two digits field, indicating the number of the concatenated parameters (the number is right justified, with leading zero if necessary).

Then follow the names of the included request parameters. A comma follows after each parameter name as a separator. The order of the parameter names indicates the order of the parameter values in the concatenation string.

For each concatenated parameter there is a 3 digits field, indicating the length of its value in the input string (the number is right justified, with leading zero if necessary):

Note: The length of the fields should be correctly calculated regardless of the encoding or char set of the string that they contain. When the string contains single-byte and multi-byte characters, for correctly calculating the length of the string it should be used functions that can handle multi-byte characters, and the length of each character should be count 1.

Header:

NN	**ParamName1,**	**ParamName2,**	…	**ParamNameNN,**	**LLL$_1$**	**LLL$_2$**	…	**LLL$_{nn}$**

For example:

08PaymentOKURL,PaymentFailURL,AmountToPay,AmountCurrency,PayToMerchant,Details1,Details2,MerchantName,025027005003010017011009

Note: the merchant checksum authentication key is not included when counting the number of parameters NN (the first two bytes of the header)

The header is used in two places:

- the value of the ChecksumHeader and the ReturnChecksumHeader request parameters

- participates in the input string for the Checksum and the ReturnChecksum request parameters

Example of Correctly Generated CheckSumHeader, InputString and CheckSum

If merchant e-shop sends the following parameters to cPay Payment Portal:

PaymentOKURL	www.OKUrl.com.mk	Telephone	38977777777
PaymentFailURL	www.FailUrl.com.mk	Email	petarp@gmail.com
AmountToPay	100	Zip	1000
AmountCurrency	MKD	Address	KJP 1/2
PayToMerchant	1234567890	City	Skopje
Details1	Detali 1	Country	807
Details2	123	OriginalAmount	10
MerchantName	ImeNaTrgovecot	OriginalCurrency	EUR
FirstName	Petar	Merchant MD5 checksum auth. key	TEST_PASS
LastName	Petrevski		

The correct **CheckSumHeader, InputString and CheckSum are:**

CheckSumHeader

18PaymentOKURL,PaymentFailURL,AmountToPay,AmountCurrency,PayToMerchant,Details1,Details2,MerchantName,FirstName,LastName,Telephone,Email,Zip,Address,City,Country,OriginalAmount,OriginalCurrency,016018003003010008003014005009011016004007006003002003

InputString

18PaymentOKURL,PaymentFailURL,AmountToPay,AmountCurrency,PayToMerchant,Details1,Details2,MerchantName,FirstName,LastName,Telephone,Email,Zip,Address,City,Country,OriginalAmount,OriginalCurrency,016018003003010008003014005009011016004007006003002003www.OKUrl.com.mkwww.FailUrl.com.mk100MKD1234567890Detali 1123ImeNaTrgovecotPetarPetrevski38977777777petarp@gmail.com1000KJP 1/2Skopje80710EURTEST_PASS

CheckSum UTF8 MD5 encoded

1AEB4E68DCF02D51C54A269EC26D94DB

Example of Correctly Generated ReturnCheckSumHeader, InputString and ReturnCheckSum

In the response, cPay echoes the parameters from the request and contains additional parameters: cPayPaymentRef (if cardholder submits card data), ReturnCheckSumHeader, ReturnCheckSum:

For the above example, if cPayPaymentRef=123456, the correct **ReturnCheckSumHeader, InputString** and **ReturnCheckSum** are**:**

ReturnCheckSumHeader

Note that the places of the first and the second parameter from the merchant’s checksum header are switched in the return checksum header

19PaymentFailURL,PaymentOKURL,AmountToPay,AmountCurrency,PayToMerchant,Details1,Details
2,MerchantName,FirstName,LastName,Telephone,Email,Zip,Address,City,Country,OriginalAmount,Ori
ginalCurrency,cPayPaymentRef,0180160030030100080030140050090110160040070060030020030
06

InputString

19PaymentFailURL,PaymentOKURL,AmountToPay,AmountCurrency,PayToMerchant,Details1,Details
2,MerchantName,FirstName,LastName,Telephone,Email,Zip,Address,City,Country,OriginalAmount,Ori
ginalCurrency,cPayPaymentRef,0180160030030100080030140050090110160040070060030020030
06www.FailUrl.com.mkwww.OKUrl.com.mk100MKD1234567890Detali
1123ImeNaTrgovecotPetarPetrevski38977777777petarp@gmail.com1000KJP
1/2Skopje80710EUR123456TEST_PASS

ReturnCheckSum UTF8 MD5 encoded

97F4E18E88A48D4BAA1742164A3AFD8B

Statement: This specification is subject to change at any time in accordance with security standards and with implementing new services.