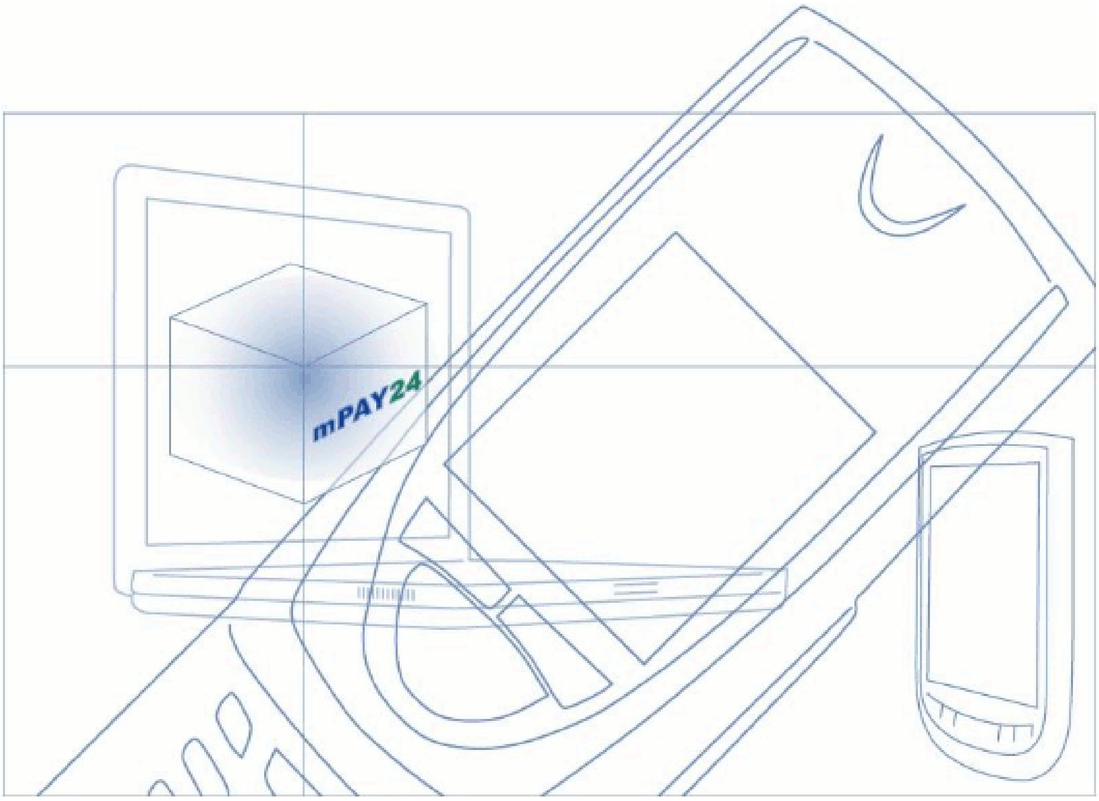
mPAY24 SOAP Specification



everything is possible

online even more!

Revision: 6130
Date: 2014-09-12

mPAY24 GmbH · Annagasse 5 · A-1010 Wien · Telefon +43-1-5136633-0 · Fax -200 · info@mPAY24.com · Sitz der Gesellschaft: Wien · FN 199939d
Handelsg: Wien · Geschäftsführer: Tom Wolf · DVR: 0832332 · UID: ATU50577503 · mPAY24 GmbH Niederlassung Berlin · Charlottenstrasse 68
D-10117 Berlin · Tel +49-30-2023 9250 · Fax +49-30-2023 9250-9 · info@mPAY24.de · Registereintrag: HRB 103190 B · Amtsgericht Charlottenburg

Contents

1.Changelog

Version	Date	Comment	Made by
-	2002-01-01	First documentation of the mPAY24 interface	-
. . .			
5564	2013-06-04	Takeover of documentation in LaTeX	sadrian
5652	2013-10-02	Base revision and interface addition	wohlmba
5672	2013-10-08	Added ip addresses and ports	wohlmba
5731	2013-10-24	Corrected parameter, added return codes, additional small changes	wohlmba
5751	2013-11-05	Enhance Confirmation Interface and SelectPayment call	wohlmba
5807	2013-12-06	Updated languages in appendix and corrected typos	wohlmba
5848	2014-01-16	Changed ELV and GIROPAY interface, added SEPA information	wohlmba
5903	2014-01-30	Added styling warning, adjusted return codes	wohlmba
5907	2014-02-05	Added implementation notes and adjusted payment system specifics	wohlmba
5961	2014-03-20	Added error/return codes	wohlmba
6040	2014-05-21	Added SOFORT Überweisung	wohlmba
6055	2014-06-05	Adjusted SOFORT Überweisungs text	wohlmba
6068	2014-07-04	Updated customerID length	wohlmba
6130	2014-09-12	Added new supported languages	wohlmba

Do not send passwords via email and state the merchant id within every support request.

Please keep all your personal and sensitive data secure. mPAY24 never requests any sensitive payment data or passwords from you. mPAY24 recommends the use of PGP (*Pretty Good Privacy*) or compatible software like GPG (e.g. *GnuPG*) for communication via E-Mail.

2.General

As payment service provider mPAY24 provides the merchant with a standardized interface to easily integrate several interfaces of financial institutions and payment systems. Within this documentation the mPAY24 interface description, implementation considerations, examples and much more are defined.

Please refer to the Appendix for general considerations and other information.

2.1. Interface update obligation

To provide new features and maintain a high security standard, mPAY24 continuously improves and updates its software. During this process it can be unavoidable to change the service interface used by the merchant. The merchant is obligated to update the implementation to the newest and safest interface of mPAY24.

Please contact the mPAY24 support for further version information and assistance during an interface change.

2.2. Reference to mPAY24

mPAY24 recommends to inform the customer that the payment will be processed by the payment service provider 'mPAY24 GmbH', e.g. by adding a short description (including the logo) of mPAY24 to the merchant's shop/web site. This increases the trustworthiness of the merchant's shop/site.

Please feel free to use the following examples:

- German:

Sichere Zahlung/Zahlungsabwicklung

Die in unserem Webshop angebotenen Zahlungsoptionen [(evtl. Aufzählung welche Zahlungssysteme verfügbar sind)] werden sicher und zuverlässig durch den mehrfach zertifizierten und ausgezeichneten Payment Service Provider (PSP) mPAY24 GmbH abgewickelt. Das Unternehmen hat sich auf Online Zahlungsverkehr spezialisiert und betreibt eine Zahlungsplattform für E- und M-Commerce.

Die mPAY24 GmbH ist zu periodischen Sicherheits-Audits im Rahmen der Zertifizierung nach dem Payment Card Industry Data Security Standard (PCI DSS) verpflichtet. Der PCI DSS umfasst umfangreiche Vorschriften im Hinblick auf technische Aspekte (Umgang mit Zahlungsdaten, Systemarchitektur, Softwaredesign) und unternehmensinterne Abläufe (Zugriffsrechte, Auskunftspolitik). Als Infrastrukturprovider unterliegt die mPAY24 GmbH der Zahlungssystemaufsicht der Österreichischen Nationalbank.

mPAY24 wurde als erster Payment Service Provider mit dem EURO-Label ausgezeichnet, erhielt bereits 2002 das Internet-Siegel des Österreichischen Handelsverbandes und hat 2005 das E-Government Gütesiegel des österreichischen Bundeskanzleramtes erhalten.

Weitere Informationen unter `https://www.mpay24.com`

- English:

Secure payment

The payment options offered in our web-shop [(optionally list of available payment systems)] are processed securely and reliably by the several times awarded and certified payment service provider (PSP) mPAY24 GmbH. The company specializes in online payments and operates a payment platform for e-and m- commerce.

mPAY24 is required to be audited to regular security audits as part of the certification according to the Payment Card Industry Data Security Standard (PCI DSS). The PCI DSS includes several recommendations and requirements related to technical aspects (handling of payment data, system architecture, software design) and internal processes (access rights, information policy). As an infrastructure provider mPAY24 GmbH is subject to the Payment Systems Oversight of the Austrian National Bank.

mPAY24 has been the first Payment Service Providers certified with the EURO- Label. In 2002 the company received the Internet Seal of the Austrian Trade Association and in 2005 the e-Government Seal of the Austrian Federal Chancellery.

More information at `https://www.mpay24.com`

3.Communication

3.1. General online transaction

Online transactions are mostly divided in two parts, the authorization/reservation and the clearing/-billing. Upon authorization, the selected payment system checks for validity, fraud prevention, coverage, etc. - if the result is positive, the requested amount is reserved for a specific time span. The actual billing is done in a separate step. The resulting advantage is that e.g. an amount smaller than the actual authorised amount can be cleared.

mPAY24 handles tasks like daily batch closing of credit card transactions automatically.

3.2. Workflow

3.2.1. General requests

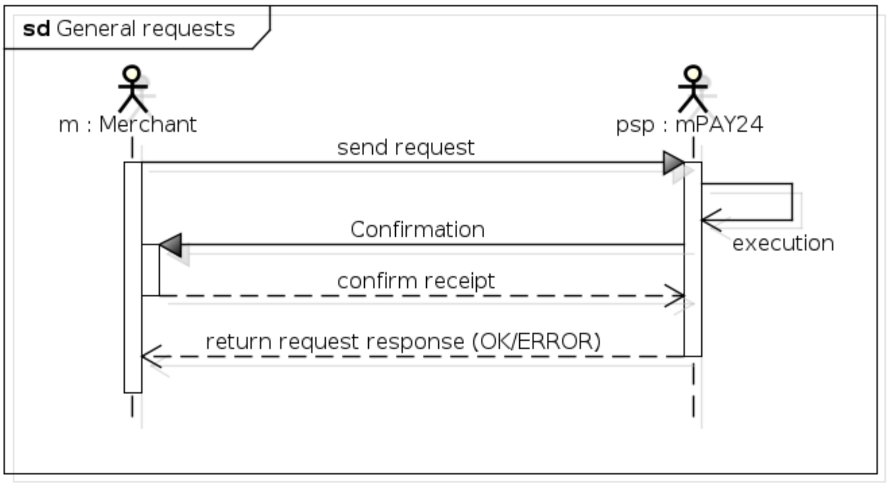


Figure 3: Workflow of general requests

1. The merchant sends a request to mPAY24 (e.g. `SelectPayment`, `ManualClear`, . . .).
2. mPAY24 processes these data and executes internal actions.
3. mPAY24 sends a Confirmation Interface call to inform the merchant about the transaction state (refer to the chapter 4 Confirmation Interface for more information). (optional step)
4. The merchant confirms the receipt of the transaction confirmation with either `OK` or `ERROR` (status depends if the confirmation could successfully update the merchant's system; refer to the chapter 4.3 Confirmation Implementation for an example). (optional step)
5. mPAY24 sends the response of the original request (including the operation `status` either `OK` or `ERROR`).

Note that calls like `DeleteProfile` or `TransactionStatus` do not change the transaction status and therefore the Confirmation Interface is not called.

It is possible that the Confirmation Interface call is received by the merchant at any time - even after the original response.

3.3. Transaction states

The following transaction states are communicated through the Confirmation Interface:

Status	Description
ERROR	The transaction failed upon the last request. (e.g. wrong/invalid data, financial reasons, ...)
RESERVED	The amount was reserved but not settled/billed yet. The transaction was successful.
BILLED	The amount was settled/billed. The transaction was successful.
REVERSED	The reserved amount was released. The transaction was cancelled.
CREDITED	The amount will be refunded. The transaction was credited.
SUSPENDED	Expecting external interface confirmation. The transaction is suspended temporarily.

3.3.1. mPAY24 PayPage

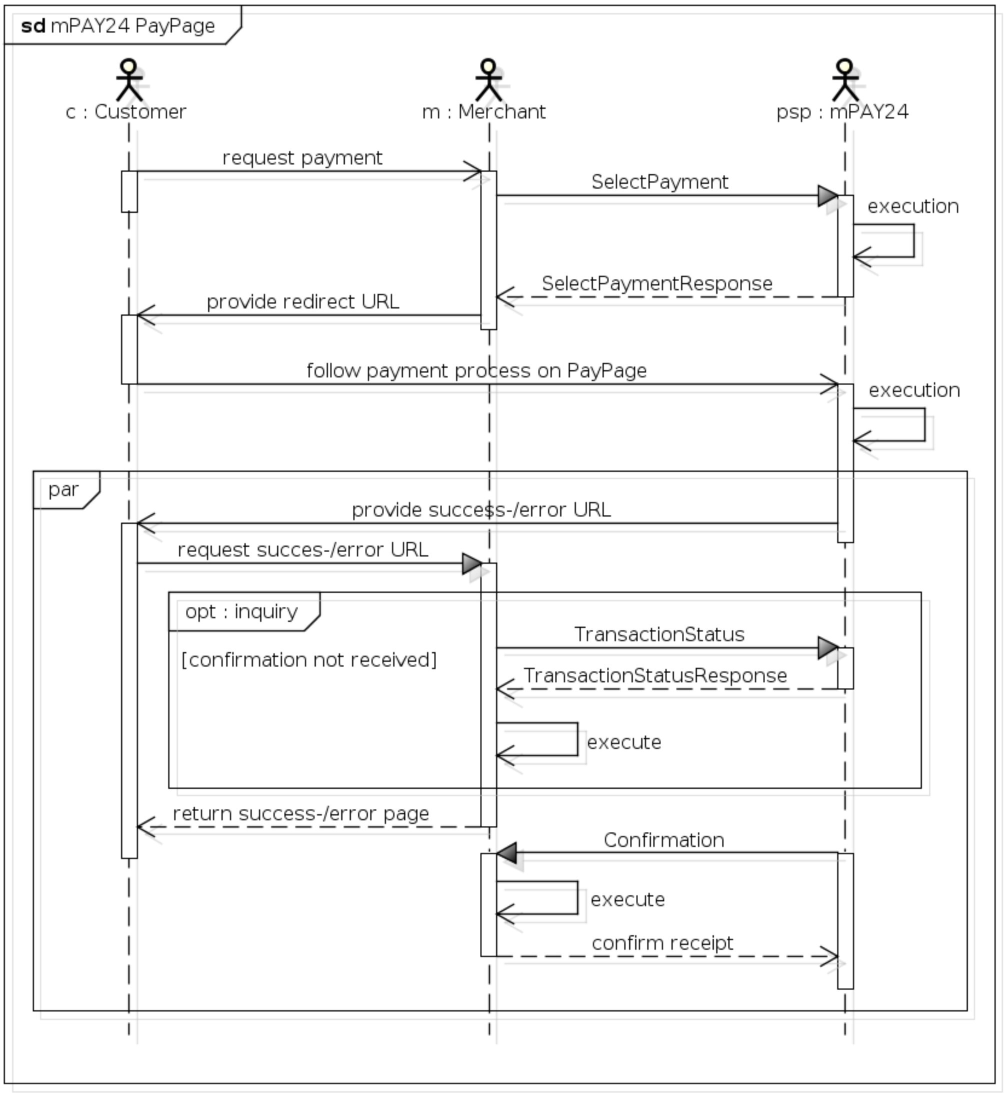


Figure 4: Workflow with the mPAY24 PayPage (SelectPayment)

1. The customer requests a payment (e.g. by pressing a *Checkout* button).
2. The merchant sends a `SelectPayment` call including details (within the MDXI XML to mPAY24).
3. mPAY24 responds with a `SelectPaymentResponse` including a `returnCode = REDIRECT` and a `location` value (if `status = OK`).
4. The customer is redirected to the mPAY24 `location` URL by the merchant (e.g. by supplying the `location` as a link or just forwarding the browser to the URL).
5. The customer follows the payment process specified by mPAY24.
6. mPAY24 processes these data and executes internal actions.
7. mPAY24 forwards the customer to the `Success-/ErrorURL` specified by the merchant dynamically within the MDXI XML or static over the Merchant Interface (MI/F).
8. The customer requests the `Success-/ErrorURL`.
9. (optional) If the Confirmation Interface has not been called yet, the merchant should initiate a `TransactionStatus` call to determine the transaction status. Never rely on the parameters provided by the `Success-/ErrorURL` alone, since they can be manipulated!
10. The merchant returns/shows the `Success-/ErrorURL` to the customer.
11. mPAY24 communicates the payment result via the Confirmation Interface.
12. The merchant confirms the receipt of the transaction confirmation with either `OK` or `ERROR` (status depends if the confirmation could successfully update the merchant's system).

4.Confirmation Interface

4.1. General

The Confirmation Interface is a proactive information interface automatically called by mPAY24. It is invoked upon a transaction status change. Please notice that, a payment transaction is identified only by the mpayTID and not by the tid assigned by the merchant. Beside the confirmation call, the merchant may also request a transaction status by calling TransactionStatus.

Notice that only up to three TransactionStatus calls are allowed for a specific transaction.

Also, if the tid is used as parameter of the TransactionStatus call, please consider that several transactions may be returned (e.g. because of multiple usage of a tid or due to customers behavior - refer to the SelectPayment call for more information).

A history of the Confirmation Interface calls can be viewed in the *Merchant Interface* (MI/F) at https://www.mpay24.com

Please notice that the confirmation interface response can only be viewed within the *Merchant Interface* (MI/F), if it is received by mPAY24 before reaching the specified time limit. The default timeout value is 20 seconds. Please contact the mPAY24 support if this value should be adjusted.

4.2. Transaction state changing

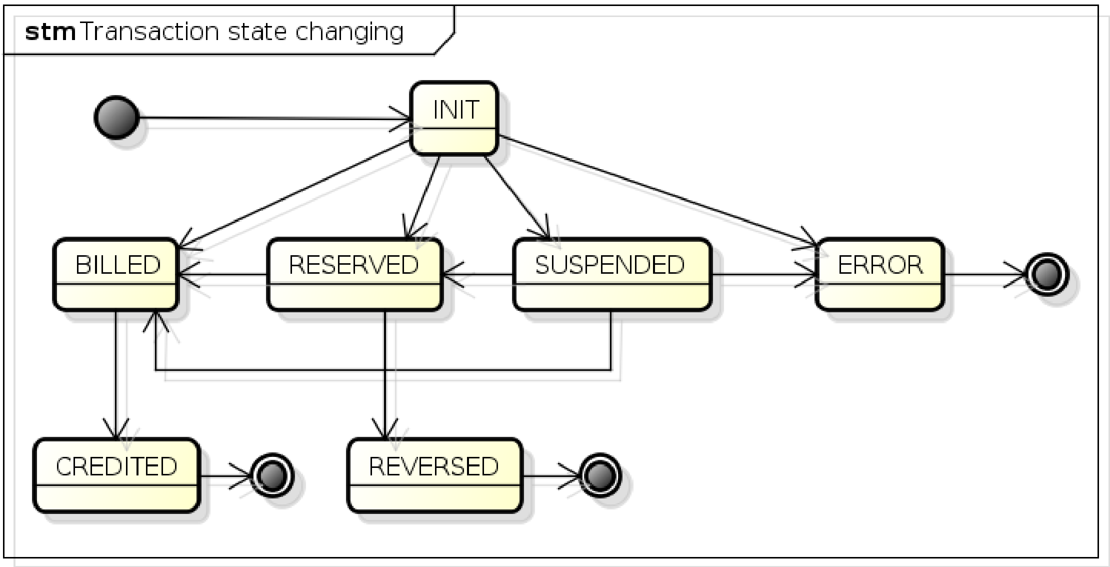


Figure 5: Showing the changing of transaction states

For further understanding, INIT is an internal status and is never returned by the Confirmation Interface as only stable states as defined in the chapter 3.3 Transaction states are returned.

Not all payment types are bound to the above status change and may invoke a Callback call from any state to any state. Currently only EPS, PAYPAL and SOFORT are able to do so.

As mentioned the Confirmation Interface is called upon a transaction state change and therefore may be called more than once for a single transaction and even months after the last confirmation call (e.g. in case of a transaction credit).

Please notice that multiple identical confirmations can be sent for a transaction (refer to chapter 4.3 Confirmation Implementation for more information).

4.3. Confirmation Implementation

The Confirmation Interface or a `TransactionStatus` call is the only possibility for a merchant to be informed about a transaction state (change) by mPAY24. Never attempt to use `successURL` or `errorURL`, as these are requested by the customer's web browser and therefore are highly unreliable and irreproducible.

As an safety measure, a daily check should be implemented, in case of transactions which were not redirected to a URL (`successURL` or `errorURL`) and did not receive a Confirmation Interface call (e.g. due to a timeout). This process should identify transactions with unknown or residual (e.g. `RESERVED` with automatically clearing enabled) transaction state within the merchant's system and initiate a `TransactionStatus` call.

4.3.1. Technical description

mPAY24 will call the `confirmationURL` of a transaction and post the information as URL parameter. The URL can be specified either dynamically within the request or static over the *Merchant Interface* (MI/F).

Example of a credit card confirmation:

```
https://www.hotelmuster.com/confirm.php?
  OPERATION=CONFIRMATION&
  TID=t121212&
  STATUS=BILLED&
  PRICE=2550&
  CURRENCY=EUR&
  P_TYPE=CC&
  BRAND=VISA&
  MPAYTID=10313717&
  USER_FIELD=My%20User%20Field&
  ORDERDESC=70000%2A2550%2AEUR&
  CUSTOMER=John%20Doe&
  CUSTOMER_EMAIL=&
  LANGUAGE=EN&
  CUSTOMER_ID=&
  PROFILE_STATUS=IGNORED&
  FILTER_STATUS=&
  APPR_CODE=123456
```

4.3.2. Default configuration

The current default configuration of the Confirmation Interface returns the following values upon a transaction change. Please contact the mPAY24 support for further information about your account configuration.

Field	Type	Length	Description
APPR_CODE	AN	≤ 80	Approval code returned by the financial institution
BRAND	AN	enum	Used brand of the payment system
CURRENCY	A	3	Used currency ISO code
CUSTOMER	AN	≤ 50	Customer name, if provided
CUSTOMER_EMAIL	AN	≤ 64	Customer e-mail address if provided
CUSTOMER_ID	AN	≤ 32	Customer ID if provided
LANGUAGE	A	2	Used language ISO code
MPAYTID	N	11	Transaction ID returned by mPAY24
ORDERDESC	AN	≤ 255	Description of the order, if not provided as default value (<merchantid>*<amount>*<currency>)
P_TYPE	AN	enum	Used payment system
PRICE	N	2 - 11	Amount of the transaction in cent
PROFILE_STATUS	AN	enum	Status of the customer profile
STATUS	AN	enum	Transaction status
TID	AN	≤ 32	Transaction ID provided by the merchant
USER_FIELD	AN	≤ 255	Content of the user field if provided

4.3.3. Securing

mPAY24 strongly advises to check the values of the received confirmation with the information known of the transaction (e.g. a transaction record on the merchant's site like a saved order of a shop system).

The following values should be checked:

- TID
- AMOUNT
- CURRENCY
- USER_FIELD

To ensure security the element UserField of the MDXI XML should be filled with a generated hash value of the tid, amount, currency, timestamp of the transaction and a random number. Additional this hash value should be saved in the merchant's shop system for comparison and checking of received confirmations.

A further security measure is to check the IP address of the confirmation call. Please refer to chapter 10 Test and productive system for the used IP addresses by mPAY24.

4.3.4. Implementation example

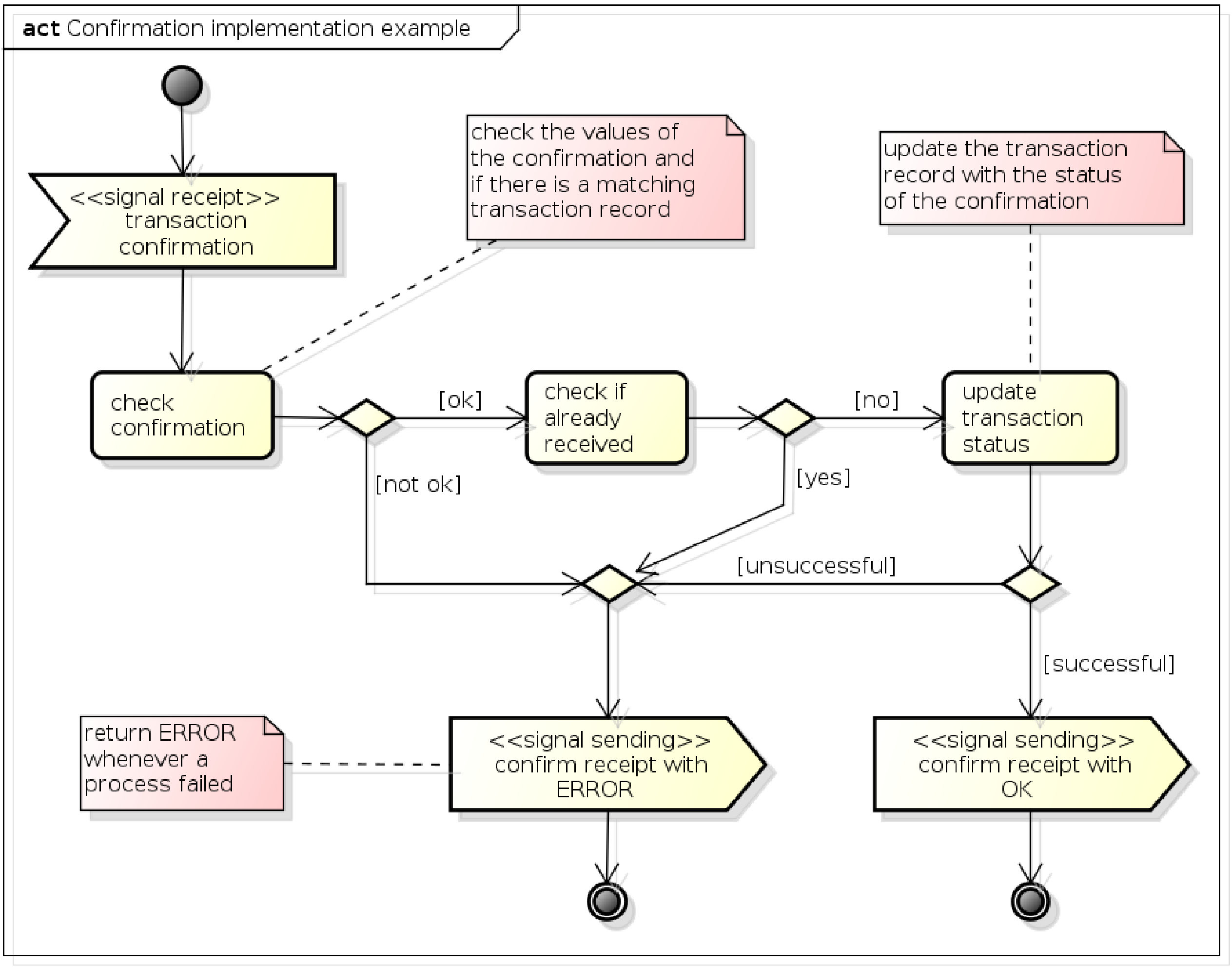

act Confirmation implementation example
<<signal receipt>> transaction confirmation
check the values of the confirmation and if there is a matching transaction record
update the transaction record with the status of the confirmation
check confirmation
[ok]
check if already received
[no]
update transaction status
[not ok]
[yes]
[unsuccessful]
[successful]
return ERROR whenever a process failed
<<signal sending>> confirm receipt with ERROR
<<signal sending>> confirm receipt with OK

Figure 6: An implementation example of the Confirmation Interface

5.SOAP Interface

5.1. Description

Webservice: ETP	
Description	mPAY24 ETP (Electronic Transaction Processing) Interface WebService
Current version	ETP v1.5
WSDL location	`https://www.mpay24.com/soap/etp/1.5/ETP.wsdl`
Endpoint prod. **Endpoint test.**	`https://www.mpay24.com/app/bin/etpproxy_v15` `https://test.mpay24.com/app/bin/etpproxy_v15`
Transport Protocol	SOAP over HTTPS
Authentication	HTTP Basic (Username/Password specified by mPAY24)
API Calls	• DeleteProfile • ListNotCleared • ListPaymentMethods • ListProfiles • ManualClear • ManualCredit • ManualReverse • SelectPayment • TransactionConfirmation • TransactionStatus All calls are of the type request-response.

5.2. Special Types

5.2.1. Address

Field	Type	Length	Description
`mode`	AddressMode	enum	Specifies whether the address can be altered by the customer upon entering the payment details
`gender`	string	enum	Customer's gender Optional, possible values: `F`, `M`
`birthday`	date	-	Customer's birthday Optional
`name`	string	≤ 50	Customer's first and last name
`street`	string	≤ 50	First street line
`street2`	string	≤ 50	Second street line Optional
`zip`	string	≤ 50	Zip code
`city`	string	≤ 50	City name
`state`	string	≤ 40	State name Optional
`countryCode`	string	2	Country as ISO code

`email`	string	$\leq$ 64	E-Mail address of the customer Optional
`phone`	string	$\leq$ 20	Phone number of the customer Optional

5.2.2. ClearingDetails

Field	Type	Length	Description
`mpayTID`	unsignedLong	$\leq$ 11	Unique transaction ID assigned by mPAY24
`amount`	unsignedInt	2 - 11	Amount to clear in cent amount Optional
`order`	Order	-	Customer order definition Optional

5.2.3. Confirmation

Field	Type	Length	Description
`state`	ConfirmationStatus	enum	Confirmation status
`confirmed`	Confirmed	enum	Confirmed transaction status
`timeStamp`	dateTime	-	Timestamp of the confirmation receipt
`url`	anyURI	-	ConfirmationURL called by mPAY24
`result`	string	$\leq$ 255	Merchant's response of the confirmation receipt

5.2.4. Item

Field	Type	Length	Description
`number`	string	$\leq$ 255	Numbers the rows of the shopping cart Optional
`productNr`	string	$\leq$ 255	Product number for the item Optional
`description`	string	$\leq$ 255	Item description respectively title Optional
`package`	string	$\leq$ 255	Scale size of the item Optional
`quantity`	unsignedInt	-	Quantity of the item ordered Optional
`tax`	unsignedInt	-	Tax of all units as cent amount Optional
`amount`	unsignedInt	2 - 11	Price of all units as cent amount

5.2.5. Order

Field	Type	Length	Description
description	string	≤ 255	Description of the order Optional
userField	string	≤ 100	Is communicated over the Success-, Error- and ConfirmationURL as USER_FIELD Optional
shoppingCart	ShoppingCart	-	Shopping cart of the order Optional
industrySpecific	IndustrySpecific	-	Common industry specific parameters Optional
billing	Address	-	Customer address definition Optional
shipping	Address	-	Customer address definition Optional

5.2.6. Parameter

Field	Type	Length	Description
name	string	≤ 255	Parameter name
value	string	≤ 255	Parameter value

5.2.7. PaymentMethod

Field	Type	Length	Description
id	unsignedInt	-	Id of the payment type
pType	PaymentType	enum	Payment type abbreviation
brand	string	≤ 25	Brand abbreviation, see chapter 7 Payment systems
description	string	≤ 80	Description of the payment system

5.2.8. Profile

Field	Type	Length	Description
customerID	string	≤ 32	Customer ID (used by mPAY24 proSAFE)
updated	dateTime	-	Timestamp of the last update

5.2.9. ShoppingCart

Field	Type	Length	Description
item	Item	-	Items in the shopping cart Optional, unbounded
discount	int	2 - 11	Discount of the total amount Optional

`shippingCosts`	int	2 - 11	Shipping costs Optional
`shippingTax`	int	2 - 11	Tax of the shipping costs Optional
`tax`	int	2 - 11	Tax of the total amount Optional

5.2.10. Transaction

Field	Type	Length	Description
`mpayTID`	unsignedLong	≤ 11	Unique transaction ID assigned by mPAY24
`tStatus`	TStatus	enum	Transaction Status
`tid`	string	≤ 32	Transaction ID given by the merchant

5.2.11. TransactionDetails

Extends the Transaction type with the following elements:

Field	Type	Length	Description
`orderDescription`	string	≤ 255	Order description Optional
`pType`	string	≤ 32	Payment type abbreviation, see chapter 7 Payment systems
`brand`	string	≤ 35	Brand abbreviation, see chapter 7 Payment systems
`amount`	unsignedInt	2 - 11	Order amount in cent
`currency`	string	3	Three digit ISO currency code
`bifStatus`	string	≤ 80	Status returned by the payment interface

5.3. Enums

5.3.1. AddressMode

Defines the mode of the address form

```
type string with restriction
     enum {'READONLY', 'READWRITE'}
```

5.3.2. ConfirmationStatus

Defines the Confirmation Interface state for transactions of `TransactionConfirmation`

```
type string with restriction
     enum {'OK', 'ERROR', 'PENDING', 'FAILURE'}
```

Notice that a confirmation results in the state `FAILURE`, if the confirmation response has been invalid.

5.3.3. Confirmed

Defines possible states returned by the Confirmation Interface after a transaction

```
type string with restriction
     enum {'RESERVED', 'BILLED', 'REVERSED', 'CREDITED', 'SUSPENDED', 'ERROR'}
```

5.3.4. PaymentType

Defines the available payment types

```
type string with restriction
     enum {'CC', 'CB', 'MAESTRO', 'EPS', 'PB', 'PSC', 'ELV', 'QUICK',
           'GIROPAY', 'PAYPAL', 'MPASS', 'BILLPAY', 'INVOICE', 'HP',
           'SAFETYPAY', 'KLARNA', 'PROFILE'}, length <= 25 chars
```

5.3.5. SortField

Defines the sort field for listing transactions

```
type string with restriction
     enum {'MPAYTID', 'PTYPE', 'BRAND', 'CURRENCY', 'TID', 'ORDERDESC',
           'ORDERNUMBER', 'CURRDATE', 'CURRTIME'}
```

5.3.6. SortType

Defines the sort type for listing transactions (ascending/descending)

```
type string with restriction
     enum {'ASC', 'DESC'}
```

5.3.7. Status

Defines possible states of a request result (does not describe the state of the transaction itself, e.g. if the transaction was successful or not)

```
type string with restriction
     enum {'OK', 'ERROR'}
```

5.3.8. TStatus

Defines possible states of a transaction result, e.g. if the transaction was successful or not

```
type string with restriction
     enum {'NOTFOUND', 'FAILED', 'RESERVED', 'BILLED', 'REVERSED', 'CREDITED'}
```

5.4. API calls

5.4.1. DeleteProfile

Deletes a stored payment profile with a specific customerID.

Input: DeleteProfile

Field	Type	Length	Description
merchantID	unsignedInt	5	Merchant ID assigned by mPAY24
customerID	string	≤ 32	Customer ID of the payment profile

Output: DeleteProfileResponse

Field	Type	Length	Description
status	Status	enum	Response status
returnCode	string	≤ 255	Return code of the request Possible values: OK, PROFILE_NOT_FOUND, INTERNAL_ERROR

5.4.2. ListNotCleared

Lists all transactions to be cleared (transactions currently residing in the status RESERVED). The result size can be restricted and sorted.

Please note, that maximal 500 transactions can be listed at once.

Input: ListNotCleared

Field	Type	Length	Description
merchantID	unsignedInt	5	Merchant ID assigned by mPAY24
begin	unsignedInt	≤ 6	Index of the first element being listed Optional, default: 0
size	unsignedInt	≤ 3	Size of returned list Optional, default: 10
sortField	SortField	enum	Specifies the parameter for ordering Optional, default: MPAYTID
sortType	SortType	enum	Specifies how elements should be ordered Optional, default: DESC
listInProgress	boolean	-	Enable/disable listing of currently processing transactions Optional

Output: ListNotClearedResponse

Field	Type	Length	Description
status	Status	enum	Response status
returnCode	string	≤ 255	Return code of the request Possible values: OK, INTERNAL_ERROR
transactionDetails	TransactionDetails	-	Detailed transaction status information Optional, unbounded

all	unsignedInt	-	Total amount of not cleared transactions

5.4.3. ListPaymentMethods

Lists all available payment methods of a merchant.
The result can be restricted to a specific PaymentType, e.g. CC for credit cards.

Input: ListPaymentMethods			
Field	**Type**	**Length**	**Description**
merchantID	unsignedInt	5	Merchant ID assigned by mPAY24
pType	PaymentType	enum	Restrict result to a specific payment type, see chapter 7 Payment systems Optional

Output: ListPaymentMethodsResponse			
Field	**Type**	**Length**	**Description**
status	Status	enum	Response status
returnCode	string	$\leq$ 255	Return code of the request Possible values: OK, INTERNAL_ERROR
paymentMethod	PaymentMethod	-	Configured Payment method Optional, unbounded
all	unsignedInt	-	Total amount of payment methods

5.4.4. ListProfiles

Lists all stored payment profiles of a merchant. The result size can be restricted.

Please note, that maximal 500 profiles can be listed at once.

Input: ListProfiles			
Field	**Type**	**Length**	**Description**
merchantID	unsignedInt	5	Merchant ID assigned by mPAY24
begin	unsignedInt	$\leq$ 6	Index of the first element being listed Optional, default: 0
size	unsignedInt	$\leq$ 3	Size of returned list Optional, default: 10

Output: ListProfilesResponse			
Field	**Type**	**Length**	**Description**
status	Status	enum	Response status
returnCode	string	$\leq$ 255	Return code of the request Possible values: OK, INTERNAL_ERROR
profile	Profile	-	Customer payment profile Optional, unbounded
all	unsignedInt	-	Total amount of stored profiles

5.4.5. ManualClear

Clears one or more transactions. Transactions have to reside in the state RESERVED.
Use ListNotCleared to determine such transactions.

Input: ManualClear			
Field	**Type**	**Length**	**Description**
merchantID	unsignedInt	5	Merchant ID assigned by mPAY24
clearingDetails	ClearingDetails	-	Clearing transaction details Unbounded

Output: ManualClearResponse			
Field	**Type**	**Length**	**Description**
status	Status	enum	Response status
returnCode	string	≤ 255	Return code of the request Possible values: OK, EXTERNAL_ERROR, INTERNAL_ERROR
transaction	Transaction	-	Transaction status information Optional, unbounded

5.4.6. ManualCredit

Credits an already cleared transaction. Transaction has to reside in the state BILLED.

Input: ManualCredit			
Field	**Type**	**Length**	**Description**
merchantID	unsignedInt	5	Merchant ID assigned by mPAY24
mpayTID	unsignedLong	≤ 11	Unique transaction ID assigned by mPAY24
amount	unsignedInt	2 - 11	Amount for clearing provided in cents Optional

Output: ManualCreditResponse			
Field	**Type**	**Length**	**Description**
status	Status	enum	Response status
returnCode	string	≤ 255	Return code of the request Possible values: OK, DECLINED, EXTERNAL_ERROR, INTERNAL_ERROR, NOT_FOUND, CACHE_DATA_EXPIRED
transaction	Transaction	-	Transaction status information Optional, unbounded

5.4.7. ManualReverse

Reverses a not cleared transaction. Transaction has to reside in the state RESERVED.

Input: `ManualReverse`			
Field	**Type**	**Length**	**Description**
`merchantID`	unsignedInt	5	Merchant ID assigned by mPAY24
`mpayTID`	unsignedLong	≤ 11	Unique transaction ID assigned by mPAY24

Output: `ManualReverseResponse`			
Field	**Type**	**Length**	**Description**
`status`	Status	enum	Response status
`returnCode`	string	≤ 255	Return code of the request Possible values: `OK`, `DECLINED`, `EXTERNAL_ERROR`, `INTERNAL_ERROR`, `NOT_FOUND`, `CACHE_DATA_EXPIRED`
`transaction`	Transaction	-	Transaction status information Optional, unbounded

5.4.8. SelectPayment

Creates an interactive payment interface (mPAY24 PayPage) for the customer to enter the payment details and perform the transaction. All payment relevant information (e.g. amount, customer name, shopping cart, ...) are provided within the `mdxi` parameter as MDXI XML.

Input: `SelectPayment`			
Field	**Type**	**Length**	**Description**
`merchantID`	unsignedInt	5	Merchant ID assigned by mPAY24
`mdxi`	string	-	XML based description, see chapter 6 MDXI XML

Output: `SelectPaymentResponse`			
Field	**Type**	**Length**	**Description**
`status`	Status	enum	Response status
`returnCode`	string	≤ 255	Return code of the request Possible values: `OK`, `INVALID_MDXI`, `REDIRECT`, `INTERNAL_ERROR`
`errNo`	int	≤ 5	Holds further data concerning an error Optional
`errText`	string	≤ 255	Holds further data concerning an error Optional
`location`	anyURI	-	URL which should be communicated to the customer's browser

By using the `SelectPayment` call, a final transaction result is depending on the customer input and behavior (e.g. the transaction was successful or not). It is also possible, that a customer initiates several transactions (with the same `tid`, since the MDXI XML description is the same during the tries). In this case either all made transactions are unsuccessful or only the last one is successful (the mPAY24 PayPage sessions terminates after a successful transaction).

Example: The merchant initiates a SelectPayment call with a MDXI description (containing: tid: order1, price: 100)).

A customer is redirected to the mPAY24 PayPage and enters his payment details, but the transaction is declined (e.g. limit exceeded). He clicks the 'Back' button on the page and tries another payment system. This time the transaction is successful and he will be redirected to the successURL.

During his purchase process he generated two different transactions:

- the first transaction, which was declined

  mpayTID: 100200, tid: order1, price: 100

- the second transaction, which was successful

  mpayTID: 100201, tid: order1, price: 100

Please note that certain payment systems require the billing and/or shipping address and/or the shopping cart. If not stated, the request can be rejected by the external interface.

mPAY24 recommends to always state as much information as possible when submitting a request.

Among others the following payment systems require additional information: ELV, BILLPAY, KLARNA

For more information about payment specific requirements please read chapter 7 Payment systems.

5.4.9. TransactionConfirmation

Returns the confirmation of a transaction specified by the mpayTID. See chapter 4 Confirmation Interface for more information.

Input: TransactionConfirmation			
Field	**Type**	**Length**	**Description**
merchantID	unsignedInt	5	Merchant ID assigned by mPAY24
mpayTID	unsignedLong	≤ 11	Unique transaction ID assigned by mPAY24

Output: TransactionConfirmationResponse			
Field	**Type**	**Length**	**Description**
status	Status	enum	Response status
returnCode	string	≤ 255	Return code of the request Possible values: OK, NOT_FOUND, INTERNAL_ERROR
confirmation	Confirmation	-	Results of a confirmationURL call Optional, unbounded

5.4.10. TransactionStatus

Returns the transaction status for a specified mpayTID or tid.

Input: TransactionStatus			
Field	**Type**	**Length**	**Description**

merchantID	unsignedInt	5	Merchant ID assigned by mPAY24
mpayTID	unsignedLong	≤ 11	Unique transaction ID assigned by mPAY24 Optional
tid	string	≤ 32	Transaction ID given by the merchant Optional

Output: TransactionStatusResponse			
Field	**Type**	**Length**	**Description**
status	Status	enum	Response status
returnCode	string	≤ 255	Return code of the request Possible values: OK, NOT_FOUND, INTERNAL_ERROR
parameter	Parameter	-	Basic name/value parameter Optional, unbounded

The transaction status can either be requested by providing the mpayTID or the tid of the transaction. If both are provided, the tid is ignored. If more than one transactions with the same tid are found, the system returns the transaction with the most recent timestamp.

Notice that only up to three TransactionStatus calls are allowed for a specific transaction. Repeating the same call in short terms of time could result in account and/or IP blocking through our security monitoring. Please use the Confirmation Interface instead to update your system on transaction status changes.

As additional security measure for the merchant, TransactionStatus should be called when the customer returns to the successURL and no Confirmation Interface status update was performed before (see chapter 4.3 Confirmation Implementation for more information).

6. MDXI XML

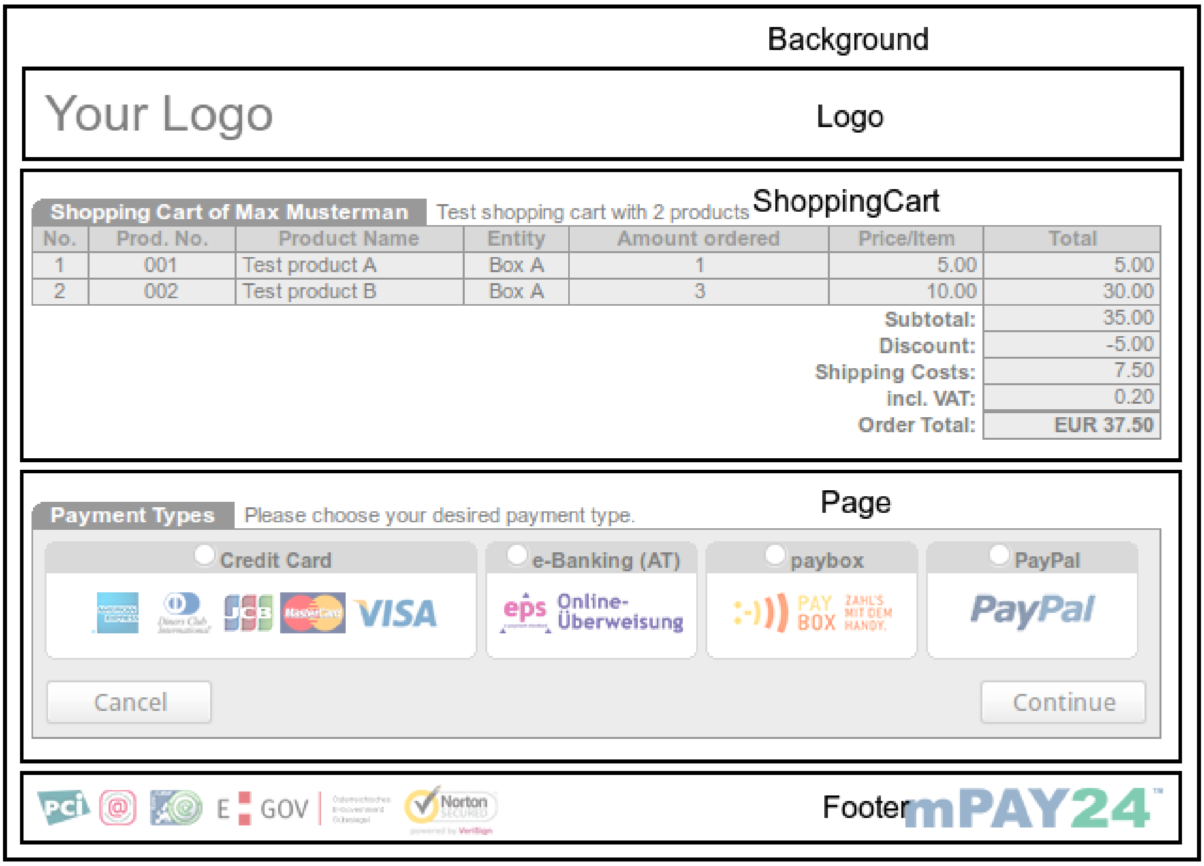
6.1. Description

The term MDXI stands for *Merchant Data Exchange Interface* and describes how the mPAY24 PayPage is constructed. It defines the shown shopping cart and the usable payment systems for the customer.

Information stated within the MDXI are passed to the financial institution, if possible. Therefore, it is recommended to provide all available information.

6.2. Customization

The mPAY24 PayPage can be adjusted by setting attributes within the MDXI. Attributes defining styles have the surfix `Style` in their name. The PayPage is divided into five parts:



Shopping Cart of Max Musterman Test shopping cart with 2 products

No.	Prod. No.	Product Name	Entity	Amount ordered	Price/Item	Total
1	001	Test product A	Box A	1	5.00	5.00
2	002	Test product B	Box A	3	10.00	30.00
					Subtotal:	35.00
					Discount:	-5.00
					Shipping Costs:	7.50
					incl. VAT:	0.20
					Order Total:	EUR 37.50

Figure 7: Description of the mPAY24 PayPage areas

Area	Description
Logo	The merchant's logo may be situated here. It has to be sent to mPAY24 support by E-Mail. The recommended size is `100x60` pixels.
ShoppingCart	Displays the content of the customer's shopping cart used for this payment.
Page	The *Page* shows the payment system selection and holds the forms where the customer can input payment specific data. It also contains information about the payment process and workflow describing messages.

Footer	The *Footer* of the page shows mPAY24 logos and certificates. Care should be taken when styling the footer area as it contains important information about security standards used at mPAY24 for the customer.
Background	The fifth part is the page background. It can be changed by placing the definitions within the attribute named `Style` in the `Order` element.

It is forbidden to refer external content within the styling attributes (e.g. an external hosted image). Referring external content can result in rejection of the request without further notice.

6.2.1. Examples

6.2.1.1. Center mPAY24 PayPage

To center the mPAY24 PayPage in the browser's display area set the `Style` attribute within the `Order` element to the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<Order Style=“margin-left: auto; margin-right: auto; width:600px;“ >
  <Tid>Tid</Tid>
  <Price>0.10</Price>
</Order>
```

6.2.1.2. Alternating item rows

To create alternating item rows the `Style` attribute has to be set for each sub element of each `Item`. Due to clarity the following code example is only an excerpt of the total MDXI:

```
<Item>
  <ProductNr Style="background-color:lightgray">Product A</ProductNr>
  <Description Style="background-color:lightgray">Desc A</Description>
  <Package Style="background-color:lightgray">Package A</Package>
  <Quantity Style="background-color:lightgray">1</Quantity>
  <ItemPrice Style="background-color:lightgray">97.50</ItemPrice>
  <Price Style="background-color:lightgray">97.50</Price>
</Item>
<Item>
  <ProductNr Style="background-color:gray">Product B</ProductNr>
  <Description Style="background-color:gray">Desc B</Description>
  <Package Style="background-color:gray">Package B</Package>
  <Quantity Style="background-color:gray">3</Quantity>
  <ItemPrice Style="background-color:gray">0.50</ItemPrice>
  <Price Style="background-color:gray">1.50</Price>
</Item>
<Item>
  <ProductNr Style="background-color:lightgray">Product C</ProductNr>
  <Description Style="background-color:lightgray">Desc C</Description>
  <Package Style="background-color:lightgray">Package C</Package>
  <Quantity Style="background-color:lightgray">4</Quantity>
  <ItemPrice Style="background-color:lightgray">0.25</ItemPrice>
  <Price Style="background-color:lightgray">1.00</Price>
</Item>
```

6.2.1.3. Mobile template

To use the mobile template, the attribute `Name` of the `TemplateSet` element have to be set to `MOBILE` – the default value would be `WEB`.

```
<?xml version="1.0" encoding="UTF-8"?>
<Order>
  <Tid>MobileTid</Tid>
  <TemplateSet Language="EN" CSSName="MOBILE"/>
  <ShoppingCart>
    <Item>
      <Description>Deluxe product</Description>
      <ItemPrice>17.00</ItemPrice>
      <Price>17.00</Price>
    </Item>
  </ShoppingCart>
  <Price>17.00</Price>
  <URL> ... </URL>
</Order>
```

It is advised to use only a selection of details for displaying the items of the shopping cart as in the above example. Otherwise the shopping cart could cause display problems.

6.3. Encoding

The encoding of the MDXI XML is either `UTF-8` or `ISO-8859-1`.

Be aware that double encoding will lead to an error and rejecting of the transaction.

6.4. Schema

The XML has to validate successful against the schema which can be obtained from:

https://www.mpay24.com/schemas/MDXI/v3.0/MDXI.xsd

The elements are documented in the order they appear in the schema.

6.4.1. Order

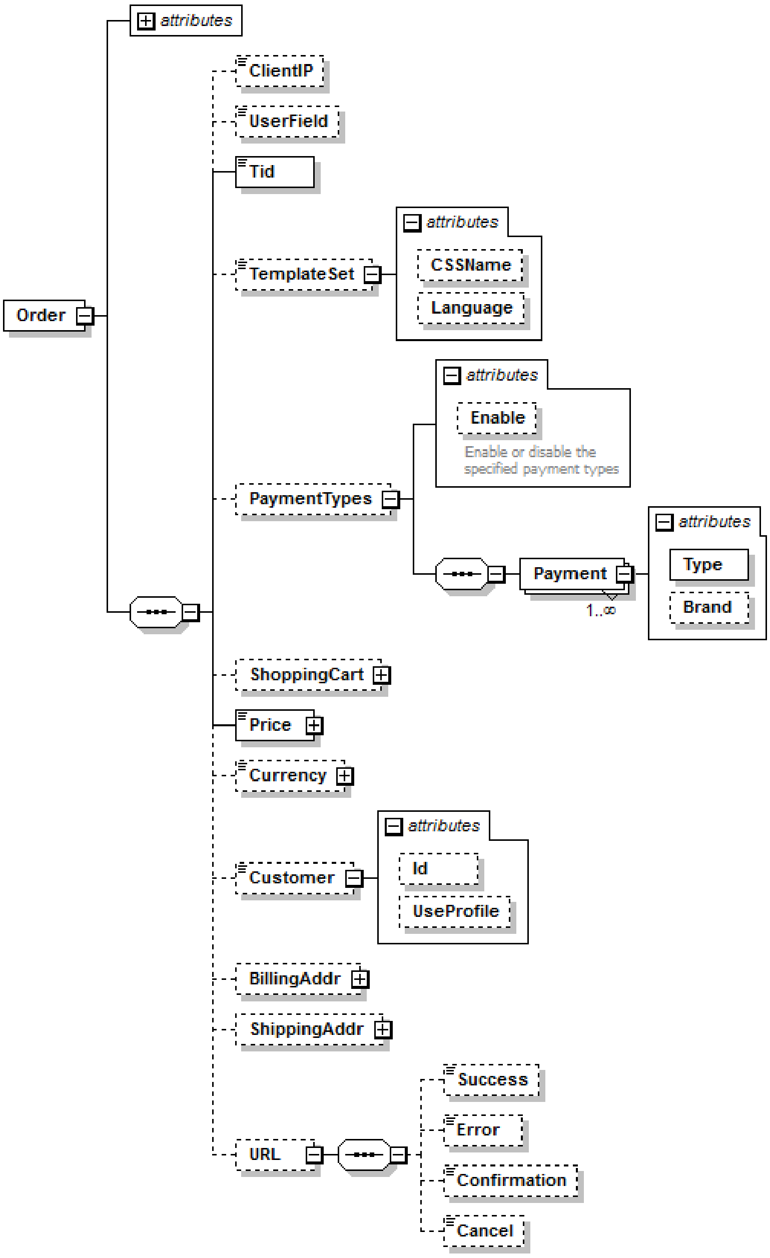


Figure 8: MDXI XML schema illustration 1

Name	Type	Length	Description
`Order`	complex	-	Root element of the MDXI XML document Sub elements: `ClientIP`, `UserField`, `Tid`, `TemplateSet`, `PaymentTypes`, `ShoppingCart`, `Price`, `Currency`, `Customer`, `BillingAddr`, `ShippingAddr`, `URL`

Attributes: `Order`			
Name	**Type**	**Length**	**Description**
`Style`	string	-	Sets the style of the *Background* area Optional

`LogoStyle`	string	-	Sets the style of the *Logo* area Optional
`PageHeaderStyle`	string	-	Sets the style of the header of the *Page* area Optional
`PageCaptionStyle`	string	-	Sets the style of the inner *Page* area Optional
`PageStyle`	string	-	Sets the style of the *Page* area Optional
`InputFieldsStyle`	string	-	Sets the style of input fields Optional
`DropDownListsStyle`	string	-	Sets the style of drop down lists Optional
`ButtonsStyle`	string	-	Sets the style of buttons Optional
`ErrorsStyle`	string	-	Sets the style of error texts Optional
`ErrorsHeaderStyle`	string	-	Sets the style of error heading texts Optional
`SuccessTitleStyle`	string	-	Sets the style of title area (mPAY24 PayPage success page) Optional
`ErrorTitleStyle`	string	-	Sets the style of title area (mPAY24 PayPage error page) Optional
`FooterStyle`	string	-	Sets the style of the *Footer* area Optional

6.4.2. ClientIP

Name	Type	Length	Description
`ClientIP`	string	≤ 15	Optional, IPv4 format Example: `208.67.222.222`

6.4.3. UserField

Name	Type	Length	Description
`UserField`	string	≤ 255	Is communicated over the `Success-`, `Error-` and `ConfirmationURL` as `USER_FIELD` Optional

6.4.4. Tid

Name	Type	Length	Description
`Tid`	string	≤ 32	Transaction ID assigned by the merchant and communicated over the `Success-`, `Error-` and `ConfirmationURL` as `TID`

6.4.5. TemplateSet

Name	Type	Length	Description
`TemplateSet`	string	≤ 255	Used to request different predefined templates Optional, possible values: `WEB` (default)

Attributes: `TemplateSet`			
Name	**Type**	**Length**	**Description**
`CSSName`	string	enum	Used to predefine a basic template Optional, possible values: `DEFAULT`, `MOBILE`
`Language`	string	2	Two digit ISO language code Optional

6.4.6. PaymentTypes

Name	Type	Length	Description
`PaymentTypes`	complex	-	Used to predefine the payment systems available to the customer for payment (black-/whitelist payment systems) Optional, sub elements: `Payment`

Attributes: `PaymentTypes`			
Name	**Type**	**Length**	**Description**
`Enable`	boolean	-	Enables/disables the payment types/brands listed in the sub elements `Payment`

The element `PaymentTypes` becomes especially useful if payment systems are to be excluded for selection by the customer. For example due to currency restrictions, payment types preselections, disabling or enabling payment methods on a per billing address country basis, . . .

Be careful when using this element, as enabling only payment systems which are not available to the merchant or disabling all payment systems available to the merchant will result in an error message upon opening the mPAY24 PayPage.

> If predefining of payment systems is unnecessary, mPAY24 recommends to not use this element, in order to use future activated payment systems immediately.

6.4.7. Payment

Name	Type	Length	Description
`Payment`	string	-	Specifies a specific payment system with its brand Unbounded

Attributes: `Payment`			
Name	**Type**	**Length**	**Description**
`Type`	string	enum	Payment type abbreviation (if only `Type` is given, all brands of this type are affected), see chapter 7 Payment systems
`Brand`	string	enum	Brand abbreviation, do not use if `Type = Brand`, see chapter 7 Payment systems Optional

6.4.8. ShoppingCart

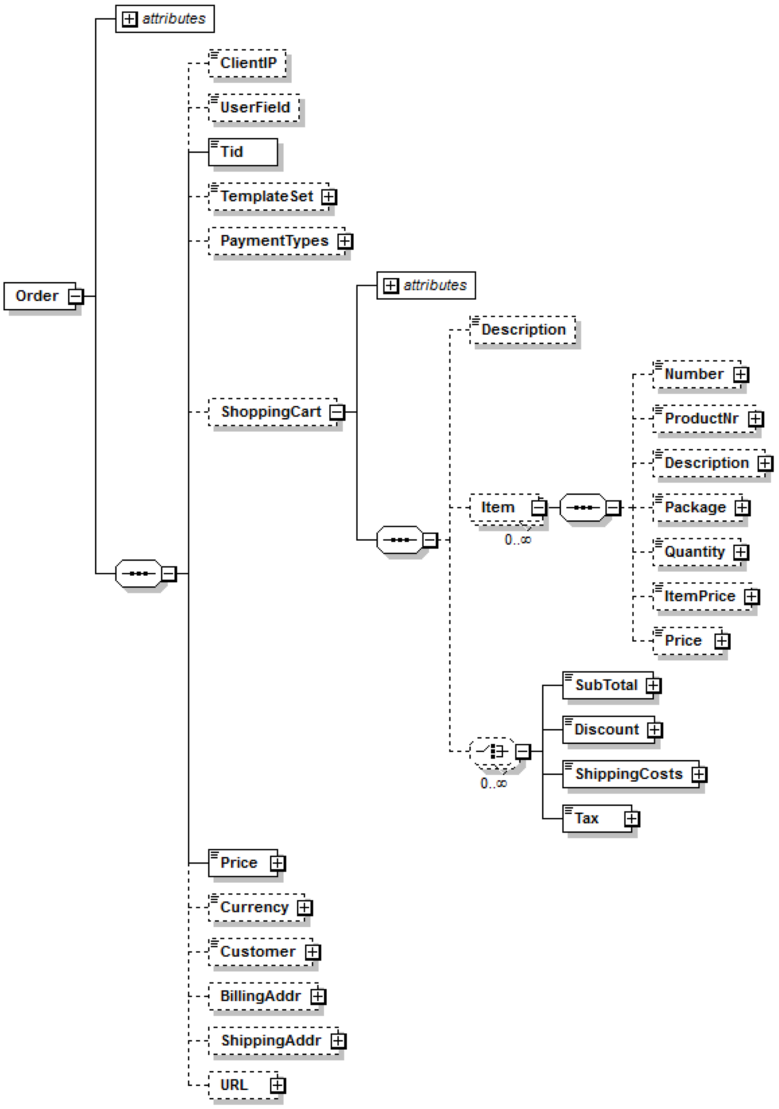


Figure 9: MDXI XML Schema illustration 2

Name	Type	Length	Description
`ShoppingCart`	complex	-	Represents the customers shopping cart Optional, sub elements: `Description`, `Item`, `SubTotal`, `Discount`, `ShippingCosts`, `Tax`

Attributes: ShoppingCart			
Name	**Type**	**Length**	**Description**
Header	string	-	Sets an alt. fixed text of the header of the *ShoppingCart* area Optional
HeaderStyle	string	-	Sets the style of the header of the *ShoppingCart* area Optional
Style	string	-	Sets the style of the *ShoppingCart* area Optional
CaptionStyle	string	-	Sets the style of the inner *ShoppingCart* area Optional
NumberHeader	string	-	Sets an alt. fixed text of the Number element Optional
NumberStyle	string	-	Sets the style of the Number element Optional
ProductNrHeader	string	-	Sets an alt. fixed text of the ProductNr element Optional
ProductNrStyle	string	-	Sets the style of the ProductNr element Optional
DescriptionHeader	string	-	Sets an alt. fixed text of the Description element Optional
DescriptionStyle	string	-	Sets the style of the Description element Optional
PackageHeader	string	-	Sets an alt. fixed text of the Package element Optional
PackageStyle	string	-	Sets the style of the Package element Optional
QuantityHeader	string	-	Sets an alt. fixed text of the Quantity element Optional
QuantityStyle	string	-	Sets the style of the Quantity element Optional
ItemPriceHeader	string	-	Sets an alt. fixed text of the ItemPrice element Optional
ItemPriceStyle	string	-	Sets the style of the ItemPrice element Optional
PriceHeader	string	-	Sets an alt. fixed text of the Price element Optional
PriceStyle	string	-	Sets the style of the Price element Optional

Please notice that if attributes are used to define an alternative text, this text is fixed and is not affected by any language setting.

6.4.9. Description

Name	Type	Length	Description
Description	string	≤ 255	Contains a description which is shown above the shopping cart Optional

6.4.10. Item

Name	Type	Length	Description
Item	complex	-	Shopping cart item Optional, unbounded, sub elements: Number, ProductNr, Description, Package, Quantity, ItemPrice, Price

Sub elements: Item			
Element name	**Type**	**Length**	**Description**
Number	string	≤ 255	Used to number the rows of the shopping cart Optional, nillable (can be left empty)
ProductNr	string	≤ 255	Product number for the item Optional
Description	string	≤ 255	Item description respectively title Optional
Package	string	≤ 255	Scale size of the item Optional
Quantity	unsignedInt	-	Quantity of the item ordered Optional
ItemPrice	float	≤ 13	Price of a single unit Example: 19.99 Mandatory if Quantity > 1 or Price is not given
Price	float	≤ 13	Price of all units Example: 19.99 Optional, nillable (can be left empty, automatically calculation: Quantity * ItemPrice)

Please see chapter 6.6 Stating tax amount in shopping cart for detailed information about net and gross price stating.

Attributes: Number, ProductNr, Description, Package, Quantity, ItemPrice			
Name	**Type**	**Length**	**Description**
Style	string	-	Sets a value style Optional

Attributes: `Price`			
Name	**Type**	**Length**	**Description**
`Style`	string	-	Sets a value style Optional
`Tax`	string	-	Absolute amount of the incurred tax within the element Optional

6.4.11. SubTotal

Name	Type	Length	Description
`SubTotal`	float	≤ 13	Sub total, e.g. for amount displaying without discount or tax Example: `19.99` Optional, unbounded

Attributes: `SubTotal`			
Name	**Type**	**Length**	**Description**
`Header`	string	-	Sets an alt. fixed label text Optional
`HeaderStyle`	string	-	Sets a label style Optional
`Style`	string	-	Sets a value style Optional

6.4.12. Discount

Name	Type	Length	Description
`Discount`	float	≤ 13	Discount of the total amount, which has to be provided as negative amount Example: `-19.99` Optional, unbounded

Attributes: `Discount`			
Name	**Type**	**Length**	**Description**
`Header`	string	-	Sets an alt. fixed label text Optional
`HeaderStyle`	string	-	Sets a label style Optional
`Style`	string	-	Sets a value style Optional

6.4.13. ShippingCosts

Name	Type	Length	Description
ShippingCosts	float	≤ 13	Shipping costs of the order Example: 19.99 Optional, unbounded

Attributes: ShippingCosts			
Name	Type	Length	Description
Header	string	-	Sets an alt. fixed label text Optional
HeaderStyle	string	-	Sets a label style Optional
Style	string	-	Sets a value style Optional
Tax	float	≤ 13	Absolute amount of the incurred tax within the element Optional

6.4.14. Tax

Name	Type	Length	Description
Tax	float	≤ 13	Tax amount of the order Example: 19.99 Optional, unbounded

Attributes: Tax			
Name	Type	Length	Description
Header	string	-	Sets an alt. fixed label text Optional
HeaderStyle	string	-	Sets a label style Optional
Style	string	-	Sets a value style Optional
Percent	string	-	Sets the Tax header in the specific language to <Percent>% Ust. or inkl. <Percent>% Ust. (if Header and HeaderStyle is not set) Optional

6.4.15. Price

Name	Type	Length	Description
Price	float	≤ 13	Total amount to be authorized/billed Example: 19.99

Attributes: Price			
Name	**Type**	**Length**	**Description**
`Header`	string	-	Sets an alt. fixed label text Optional
`HeaderStyle`	string	-	Sets a label style Optional
`Style`	string	-	Sets a value style Optional

6.4.16. Currency

Name	Type	Length	Description
`Currency`	string	3	Three digit ISO currency code Optional, default: `EUR`

Attributes: Currency			
Name	**Type**	**Length**	**Description**
`Display`	string	-	Alternative value to be displayed instead of three digit ISO currency code Optional

6.4.17. Customer

Name	Type	Length	Description
`Customer`	string	$\leq$ 50	Customer is intended for usage with mPAY24 proSAFE, also used if the customer's name differs to the name specified in the billing address Optional

Attributes: Customer			
Name	**Type**	**Length**	**Description**
`Id`	string	$\leq$ 32	Customer ID assigned by the merchant Optional
`UseProfile`	boolean	-	mPAY24 proSAFE: defines if the profile should be used/stored Optional, default: `false`

6.4.18. BillingAddr

Name	Type	Length	Description
`BillingAddr`	AddressType	-	Address used for billing Optional

6.4.19. ShippingAddr

Name	Type	Length	Description
ShippingAddr	AddressType	-	Address used for shipping Optional

If ShippingAddr is not set, the same values as for BillingAddr are assumed for the shipping address.

6.4.20. AddressType

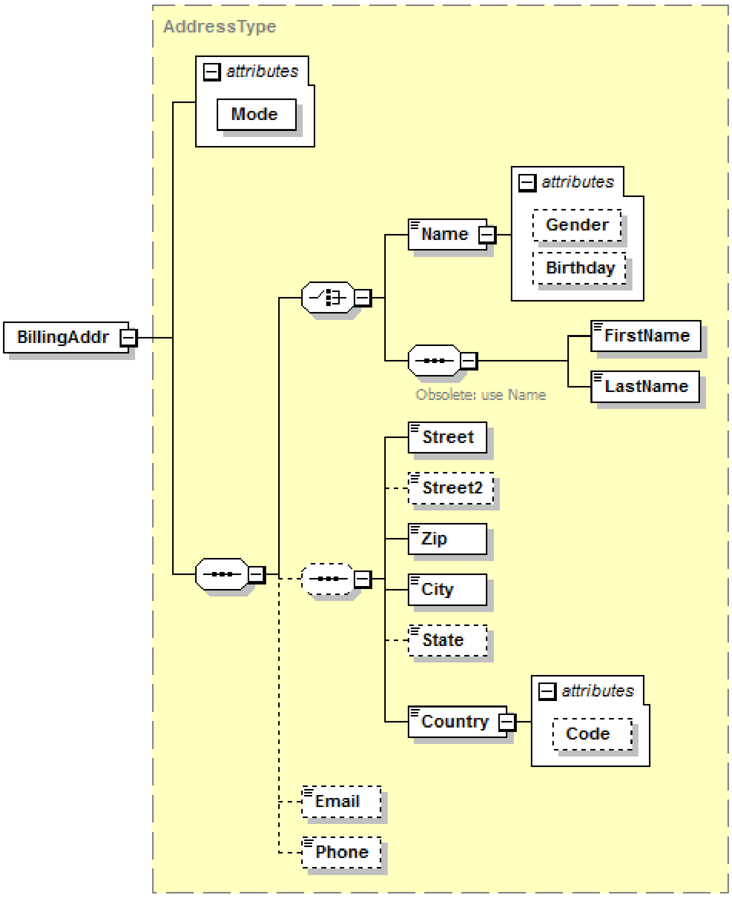


Figure 10: MDXI XML schema complex type AddressType illustration

Name	Type	Length	Description
AddressType	complex	-	Specifies an address Optional, sub elements: Name, FirstName, LastName, Street, Street2, Zip, City, State, Country, Email

Attributes: AddressType			
Name	**Type**	**Length**	**Description**
Mode	string	enum	Specifies whether the address can be altered by the customer upon entering the payment details Possible values: ReadOnly, ReadWrite

Sub elements: AddressType			
Element name	**Type**	**Length**	**Description**
Name	string	≤ 50	Customer's first and last name First and last name should be separated by a space, while e.g. multiple first names should be separated by an underscore Example: John Doe, Jean_Dora Doe
FirstName	string	-	Deprecated
LastName	string	-	Deprecated
Street	string	≤ 50	First street line
Street2	string	≤ 50	Second street line Optional
Zip	string	≤ 50	Zip code
City	string	≤ 50	City name
State	string	≤ 40	State name Optional
Country	string	≤ 50	Country name
Email	string	≤ 64	E-Mail address of the customer Optional
Phone	string	≤ 20	Phone number of the customer Optional

The AddressType requires to at least state a name. If either Street, Zip, City or State is specified, the other elements (Street, Zip, City, State) are also mandatory to specify. The elements Street2, State, Email and Phone are always optional.

Attributes: Name			
Name	**Type**	**Length**	**Description**
Gender	string	enum	Gender of the customer Optional, possible values: F, M
Birthday	date	-	Birthday of the customer Optional, format: YYYY-MM-DD Example: 1990-01-31

Attributes: `Country`			
Name	**Type**	**Length**	**Description**
`Code`	string	2	Two digit ISO country code Optional

6.4.21. URL

Name	Type	Length	Description
`URL`	complex	-	Specifies an URL Optional, sub elements: `Success`, `Error`, `Confirmation`, `Cancel`

Sub elements: `URL`			
Element name	**Type**	**Length**	**Description**
`Success`	anyURI	≤ 1024	URL, where the customer is to be redirect in case of authorization success (starts with `http://` or `https://`) Optional
`Error`	anyURI	≤ 1024	URL, where the customer is to be redirect in case of authorization error (starts with `http://` or `https://`) Optional
`Confirmation`	anyURI	≤ 1024	URL for receiving confirmation of the transaction state (starts with `http://` or `https://`) Optional
`Cancel`	anyURI	≤ 1024	URL, where the customer is to be redirect in case of cancellation (before payment/authorization) (starts with `http://` or `https://`) Optional

If the `Success-`, `Error-` or `CancelURL` is not defined within the MDXI, it is either stored in the mPAY24 system or the payment ends in the mPAY24 PayPage.

6.5. Example

```
<?xml version="1.0" encoding="UTF-8"?>
<Order>
  <ClientIP>208.67.222.222</ClientIP>
  <UserField>User Field 2012-09-04 09:33:39</UserField>
  <Tid>0000023hf9272324gd</Tid>
  <TemplateSet Language="EN">WEB</TemplateSet>
  <ShoppingCart>
    <Description>Test shopping cart with 2 products</Description>
    <Item>
      <Number>1</Number>
      <ProductNr>001</ProductNr>
      <Description>Test product A</Description>
      <Package>Box A</Package>
      <Quantity>1</Quantity>
      <ItemPrice Tax="0.20">5.00</ItemPrice>
      <Price>5.00</Price>
    </Item>
    <Item>
      <Number>2</Number>
      <ProductNr>002</ProductNr>
      <Description>Test product B</Description>
      <Package>Box A</Package>
      <Quantity>3</Quantity>
      <ItemPrice Tax="0.20">10.00</ItemPrice>
      <Price>30.00</Price>
    </Item>
    <SubTotal>35.00</SubTotal>
    <Discount>-5.00</Discount>
    <ShippingCosts>7.50</ShippingCosts>
    <Tax>0.20</Tax>
  </ShoppingCart>
  <Price>37.50</Price>
  <Currency>EUR</Currency>
  <BillingAddr Mode="ReadOnly">
    <Name Gender="M" Birthday="1990-01-31">John Doe</Name>
    <Street>Billing Street 1</Street>
    <Street2>Billing Street 2</Street2>
    <Zip>1234</Zip>
    <City>Vienna</City>
    <Country Code="AT"/>
    <Email>billing@mpay24.test</Email>
  </BillingAddr>
  <ShippingAddr Mode="ReadOnly">
    <Name Gender="F" Birthday="1992-02-20">Jane Doe</Name>
    <Street>Shipping Street 1</Street>
    <Street2>Shipping Street 2</Street2>
    <Zip>98765</Zip>
    <City>London</City>
    <Country Code="GB"/>
    <Email>shipping@mpay24.test</Email>
  </ShippingAddr>
  <URL>
    <Success>https://www.hotelmuster.com/.../success.php?lang=EN</Success>
    <Error>https://www.hotelmuster.com/.../error.php?lang=EN</Error>
    <Confirmation>https://www.hotelmuster.com/.../confirm.php?</Confirmation>
    <Cancel>https://www.hotelmuster.com/.../cancel.php?lang=EN</Cancel>
  </URL>
</Order>
```

6.6. Stating tax amount in shopping cart

Please be aware that mPAY24 does not calculate any net or gross amounts as they have to be provided within the ShoppingCart. All amounts have to be stated absolutely and not percentaged.

mPAY24 does not notice the merchant if the sum of the shopping cart and the total sum are not equal. When the sums are differing, the shopping cart will be discard and not transmitted to the financial institution.

6.6.1. Net prices

If the shopping cart should contain only net prices, the elements ItemPrice have to be defined with net amounts. The attribute Tax should not be set.
The element Tax holds the absolute value of the incurred tax of the total amount.
Therefore, the total amount of the order (element Price) is made up from each Price element within an Item and the Tax element itself.

```
<?xml version="1.0" encoding="UTF-8"?>
<Order>
  <Tid>Order excl. Tax</Tid>
  <ShoppingCart>
    <Item>
      <Number>1</Number>
      <Quantity>1</Quantity>
      <ItemPrice>0.80</ItemPrice>
    </Item>
    <Item>
      <Number>2</Number>
      <Quantity>2</Quantity>
      <ItemPrice>1.60</ItemPrice>
    </Item>
    <Tax>1</Tax>
  </ShoppingCart>
  <Price>5</Price>
  <URL> ... </URL>
</Order>
```

6.6.2. Gross prices

If the shopping cart should contain only gross prices, the element ItemPrice have to be defined with gross amounts. The attribute Tax have to be set to the absolute value of the incurred tax within the ItemPrice element.
The element Tax holds the absolute value of the incurred tax of the total amount.
The total amount of the order (element Price) is made up only from each Price element and optional from the Tax attribute of ShippingCosts, since these prices already include the tax. Additional the tax will be displayed with the abbr. "incl." prefix on the mPAY24 PayPage, if no header was set (via the Header attribute).

```
<?xml version="1.0" encoding="UTF-8"?>
<Order>
  <Tid>Order incl. Tax</Tid>
  <ShoppingCart>
    <Item>
      <Number>1</Number>
      <Quantity>1</Quantity>
      <ItemPrice Tax="0.20">1
    </ItemPrice>
    </Item>
    <Item>
      <Number>2</Number>
      <Quantity>2</Quantity>
      <ItemPrice Tax="0.40">2
    </ItemPrice>
    </Item>
    <Tax>1</Tax>
  </ShoppingCart>
  <Price>5</Price>
  <URL> ... </URL>
</Order>
```

7.Payment systems

7.1. Available payment systems

In the following table all available payment systems are listed with their names and payment type abbreviations used by mPAY24. Note that some payment systems require to redirect the customer to a third party site, e.g. a bank site.

Name	Payment Type	Brand	Redirect
Carte Bleue	CB	-	No
Credit Cards	CC	AMEX	No
		DINERS	No
		JCB	No
		MASTERCARD	Depending on variant
		VISA	Depending on variant
Direct Debit	ELV	ATOS	No
		BILLPAY	Yes
		HOBEX-AT	No
		HOBEX-DE	No
		HOBEX-NL	No
eps Online-Überweisung	EPS	EPS	Yes
giropay	GIROPAY	-	Yes
Klarna	KLARNA	HP	No
		INVOICE	No
Billpay	BILLPAY	HP	Yes
		INVOICE	Yes
Maestro SecureCode	MAESTRO	-	Yes
mpass	MPASS	-	Yes
PayPal	PAYPAL	-	Yes
paybox	PB	-	No
mPAY24 proSAFE	PROFILE	-	No
paysafecard	PSC	-	Yes
@Quick	QUICK	-	Yes
Safetypay	SAFETYPAY	-	Yes
SOFORT Überweisung	SOFORT	-	Yes

In case of successful payment initiation, almost all payment types produce a `returncode = REDIRECT`. Do not attempt to display the `location` or the mPAY24 PayPage in an IFRAME or pop-up to avoid displaying errors of third party sites.

mPAY24 recommends to implement the `location` or the mPAY24 PayPage as full screen.

7.2. Restrictions

Clearings may only be issued of amounts smaller or equal the authorized amounts. Accordingly, credits may also only be issued if the cleared amount is smaller or equally unless stated otherwise.

Credits may be possible within 1 year after clearing unless stated otherwise.

Payment types not supporting authorization or cancellation return a `status = BILLED` without returning `status = RESERVED` beforehand on the Confirmation Interface.

Payment Type	Brand	Authorization	Credit
CB	CB	14 days*	Applicable
CC	AMEX	14 days*	Only one credit
	DINERS	14 days*	Only one credit
	JCB	14 days*	Only one credit
	MASTERCARD	14 days*	Only one credit
	VISA	14 days*	Only one credit
ELV	ATOS	N/A	N/A
	BILLPAY	30 days	N/A
	HOBEX-AT**	Until 11pm the same day	Until 11pm the same day
	HOBEX-DE**	Until 11pm the same day	Until 11pm the same day
	HOBEX-NL**	Until 11pm the same day	Until 11pm the same day
EPS	EPS	N/A	N/A
GIROPAY	-	N/A	N/A
KLARNA	HP	N/A	N/A
	INVOICE	N/A	N/A
HP	-	N/A	N/A
INVOICE	-	N/A	N/A
BILLPAY	HP	30 days	Applicable
	INVOICE	30 days	Applicable
MAESTRO	-	N/A	N/A
MPASS	-	Applicable	Credit requests after 0pm the same day result in add. fees
PAYPAL	-	As defined	As defined
PB	-	Applicable	Within 90 days
PROFILE	-	Depend on the payment profile	Depend on the payment profile
PSC	-	1 hour	N/A
QUICK	-	N/A	N/A
SAFETYPAY	-	30 days	Applicable
SOFORT	-	N/A	N/A

* if no other agreement is defined
** only 3 successful authorizations during one calendar day (since 2013-02-13) and only amounts equal the authorized amount can be cleared

7.3. Additional

7.4. Further documentation

Depending on the used payment system or acquirer further implementation notes and information can be obtained of additional documents. Please refer to the following list and request the additional documents from mPAY24 if not already received.

Acquirer	Payment system	Document
Billpay	`ELV`, `HP`, `INVOICE`	mPAY24-Billpay-Specification
PayPal	`PAYPAL`	mPAY24-PayPal-API-Permissions
-	`PROFILE`	mPAY24-proSAFE

7.5. Payment system specifics

7.5.1. ELV - Direct Debit

7.5.1.1. Provide customer name (HOBEX)

Payments with `brand = HOBEX` require a customer name within the request, regardless if the merchant uses `AcceptPayment` or `SelectPayment`.

7.5.1.2. ATOS SEPA mandates

mPAY24 highly recommends the merchant to additionally request a written SEPA mandate from the customer, regardless if the merchant uses `AcceptPayment` or `SelectPayment`.

7.5.2. SOFORT - SOFORT Überweisung

SOFORT Überweisung returns the status `SUSPENDED` after the customer entered all his credentials on the SOFORT Überweisung page and the transaction was successfully set up in the customers online banking.

Please do not interprete the `SUSPENDED` status as the confirmation of a successful payment as due to the respective banks terms of business the customer may have the possibility to stop or reverse the transfer of funds.

If the merchant holds a SOFORT bank account, the status is updated as soon as the amount is received on the account.

If the merchant does not hold a Sofort bank account, the status remains in this state. That means the merchant does not receive a `BILLED` status for this transaction.

8.Return codes and error messages

Generally, there are three main types of error messages:

- DECLINED
  The transaction was declined by the specific external payment interface and can not be processed any more.
- INTERNAL_ERROR
  An error during the communication occurred.

  Please notice that this error is also returned if the external payment interface returns an incorrect or unknown error code.
- EXTERNAL_ERROR
  An error occurred while processing through the external payment interface and a correct error code was returned by this interface.

  The return value errText may contains additional information supplied by the external payment interface if a public error code (or message) was defined by the interface. If possible, the error message is returned in the language defined upon request (default: DE).

  In case of a non public error code as defined by the external payment interface, a general message will be provided, since the merchant or respectively the customer is not allowed to be notified about the reason of the error.

8.1. Return codes

Nr.	ReturnCode	Description
0	OK	The transaction was ok (no error occurred).
1	DECLINED	The transaction was declined by the external payment interface. The customer should refer to his issuer about the reason of the failure.
2	BLOCKED	The transaction was blocked
3	ACCESS_DENIED	The merchant's IP address is not white listed. Please contact the mPAY24 support.
4	MERCHANT_LOCKED	The merchant id is locked. Contact mPAY24 support for more information
5	PAYMENT_METHOD_NOT_ACTIVE	The desired payment system is not active. Please contact the mPAY24 support
6	<parameter>_NOT_ENTERED	<parameter> is replaced by a specific parameters name. A mandatory parameter is missing.
7	<parameter>_NOT_CORRECT	<parameter> is replaced by a specific parameters name. The parameter supplied is not valid (e.g only numeric values are expected).
8	<parameter>_NOT_SUPPORTED	<parameter> is replaced by a specific parameters name. The parameter is not supported.
9	PTYPE_MISMATCH	The payment method mismatches e.g. with an id or a brand.

10	NOT_FOUND	The mpayTID or tid of the transaction was not found.
11	ALREADY_PROCESSED	The transaction has already been processed and can not be processed again.
12	CACHE_DATA_EXPIRED	The temporary cache data are invalid due to expiration.
13	INVALID_MDXI	The MDXI XML stream could not be validated (refer to chapter 6 MDXI XML).
14	INVALID_AMOUNT (INVALID_PRICE)	The parameter amount/price holds invalid values.
15	INVALID_CREDITCARD_NUMBER	The credit card number is not plausible (e.g. cannot exist or contains invalid characters like spaces).
16	INVALID_MAESTRO_NUMBER	The Maestro card number is not plausible (e.g. cannot exist or contains invalid characters like spaces).
17	INVALID_IBAN	The provided IBAN is not plausible (e.g. cannot exist or contains invalid characters).
20	PROFILE_NOT_FOUND	mPAY24 payment profile could not be found.
21	PROFILE_NOT_SUPPORTED	mPAY24 proSAFE is not activated. Please contact the mPAY24 support
22	PROFILE_FLEX_NOT_SUPPORTED	mPAY24 proSAFE FLEX is not activated. Please contact the mPAY24 support
23	PROFILE_COUNT_EXCEEDED	The maximum number of payment profiles of the customer was already reached.
99	INTERNAL_ERROR	An error during the communication occurred.
100	EXTERNAL_ERROR	The external payment interface returned an error.

9.Timeouts

9.1. mPAY24 PayPage session and idle timeouts

The mPAY24 PayPage is configured for a total session timeout of 30 minutes. The session idle timeout is set to 15 minutes. If one of these timeouts is exceeded, the session is terminated and the payment should be restarted.

Notifications for successful payments could occur even if the session is expired. In such case the customer is returned to the `successURL`.

9.2. Specific payment system timeouts

The following timeout values are known from the various payment systems. Please note, that this information is subject to change without notice and does not necessarily reflect the actual setting in place.

Payment Type	Specific system name	Timeout information
`EPS`	eps Online-Überweisung	Varies depending of the financial institution
`GIROPAY`	giropay	5 minutes
`MASTERCARD`	Mastercard SecureCode	Varies depending of the financial institution
`PAYPAL`	PayPal	5 minutes as long as there is no customer interaction; no total timeout
`PB`	paybox	3 minutes
`PSC`	paysafecard	30 minutes
`VISA`	Verified by VISA	Varies depending of the financial institution (at least 5 minutes)

10.Test and productive system

mPAY24 creates an account on the productive and test system for each regular merchant. These accounts are strictly separated and passwords are generated independently.

- Production system
  - `merchantID` beginning with `7`
  - Merchant Interface (MI/F): `https://www.mpay24.com/`
  - Please note, that the account on the production system cannot be used right away as production mode has to be requested from mPAY24 Support
- Test system
  - `merchantID` beginning with `9`
  - Merchant Interface (MI/F): `https://test.mpay24.com/`

> Live transactions initiated on the test system cannot be recovered. Therefore, it is the merchant's responsibility to decide between test and productive system.

10.1. Merchant Interface (MI/F)

The Merchant Interface (MI/F) enables the merchant to view/search/download transactions.

Please note, that after three incorrect authentication attempts the access to the Merchant Interface (MI/F) is blocked for 30 min. Please try again after this time or contact the mPAY24 Support for a new password.

The Merchant Interface (MI/F) manual can be downloaded within the MI/F after successful authentication.

10.2. Changing SOAP interface into productive mode

In order to switch from the test to the productive system, the following steps are required:

1. change the authentication data accordingly (`merchantID` beginning with `7` and associated Webservice/SOAP password)
2. changing the WSDL endpoint location (see chapter 5 SOAP Interface)
   a) not necessary if the mPAY24 API is used
3. request productive mode by contacting the mPAY24 Support

If troubles occur while switching from the test to the productive system, make sure the correct Webservice/SOAP password have been used.

mPAY24 will inform you about the successful activation of your productive account. For confirmation, please log in at `https://www.mpay24.com` and check on the *Partner Details* screen, if the message *NOTE: Your account is currently in test mode.* has disappeared.

10.3. IP addresses and ports

	Merchant to mPAY24	mPAY24 to merchant
Productive system	www.mpay24.com 213.164.25.234	213.164.25.245
Test system	test.mpay24.com 213.164.23.169	213.164.23.169
Ports	443	80 or 443

Concerning the Confirmation Interface, please make sure that the access will be allowed or rather that the Confirmation Interface is not called by another IP address.

10.4. Test system

The mPAY24 test system is used by mPAY24 for release tests and therefore may not be available 24/7. mPAY24 reserves the right to disable or discontinue the test system without further notice. Support for the mPAY24 test system is only available during the office hours.

Please note that only VISA is configured on the test system by default. If you need any other payment system, please contact the mPAY24 support.

10.4.1. Test details

The following payment systems and details are available on the mPAY24 test system.

Please note that it is strictly forbidden to use productive payment details on the mPAY24 test system as well as use to use test payment details on the mPAY24 productive system.

10.4.1.1. Credit Cards

The following test credit card numbers are available on the mPAY24 test system:

CC Brand	Test number
VISA	4444333322221111
MASTERCARD	5555444433331111

Various test scenarios can be created with the expiry date of the credit card:

Parameter	Scenario	Result
CVC	Irrelevant on the test system, e.g. 123	
Expiry month	< 05	Various error messages
	= 05	Successful transaction
	> 05	3-DS (MasterCard SecureCode/Verified by VISA transaction) with possibility to choose further options on 3-DS test page

10.4.1.2. ELV (Elektronisches Lastschrift Verfahren)

The following test data can be used on the test system only for `ELV HOBEX` transactions:

```
IBAN: AT77 1400 0001 2345 6789
BIC: BAWAATWWXXX
```

10.4.1.3. EPS

Details for test mode of EPS are either supplied directly by the banks upon an eps Online-Banking agreement with them or the following test details of the Erste Bank Sparkasse can be used:

Erste Bank Sparkasse

Please note that supplying wrong login details frequently will lead to the account being locked. All costs incurred will fully be charged to the initiator.

Credentials:

```
User:  108256743
Password:  npydemo
```

TANs	Scenario result
11111	Valid TAN
22222	Various error messages
33333	
44444	
55555	

10.4.1.4. Maestro SecureCode

For expiry, the same rules as for credit cards apply, except that it is not possible to process non 3-DS transactions.

Parameter	Number
Identifier	6700555544444444

10.4.1.5. paybox

The following phone numbers can be used in test mode:

Number	Result
0664 1111 111	Immediately response
0664 2222 222	Delayed response

10.4.1.6. paysafecard

The following table contains test data for paysafecard:

#	PIN	S/N	Value	Currency
1	0000000077203985	77203985	50.00	EUR
2	0000000077303985	77303985	50.00	EUR
3	0000000077403985	77403985	50.00	EUR
4	0000000077503985	77503985	50.00	EUR
5	6363110001003985	6363110001003985	10.00	USD
6	6363110001103985	6363110001103985	25.00	USD
7	6364070100203985	6364070100203985	100.00	MXN
8	6364070100303985	6364070100303985	250.00	MXN
9	6364070100403985	6364070100403985	500.00	MXN
10	6364070100503985	6364070100503985	500.00	MXN
11	0000000799703985	799703985	5.00	USD
12	6500000002003985	6500000002003985	100.00	EUR
13	0000000799503985	799503985	30.00	CHF
14	0000000101503985	101503985	5.00	EUR
15	0000000101603985	101603985	2.50	EUR
16	0000000101703985	101703985	3.60	EUR
17	0000000101803985	101803985	4.70	EUR
18	0000000101903985	101903985	4.19	EUR
19	0000000102003985	102003985	4.20	EUR
20	0000000102103985	102103985	4.21	EUR
21	0000000102203985	102203985	5.00	EUR
22	0000000102303985	102303985	5.00	EUR
23	0000000002503985	2503985	5000.00	SIT
24	0000000002603985	2603985	100000.00	SIT
25	0000000003003985	3003985	500.00	SKK
26	0000000003103985	3103985	10000.00	SKK
27	3000000000003985	3000000000003985	100.00	EUR
28	5000000000003985	5000000000003985	51.13	EUR
29	8888888888803985	8888888888803985	500.00	EUR
30	6000000000003985	6000000000003985	51.13	EUR
31	2000000000003985	2000000000003985	36.34	EUR
32	1000000000003985	1000000000003985	100.00	EUR
33	0000000000103985	103985	21.80	EUR
34	4000000000003985	4000000000003985	36.34	EUR
35	9000000000003985	9000000000003985	36.34	EUR

10.4.1.7. Sofort Überweisung

The SOFORT Überweisungs project has to be configured as 'test project' to use the following test details.

Bank code (BIC): `00000`

After entering the test bank code, all other data are irrelevant and can be entered as one wish.

DO NOT use real data on the test environment.

11. SOAP test examples

11.1. SelectPayment with short MDXI

In this example, a payment is initiated with a minimal MDXI XML as described in the chapter 6 MDXI XML. It contains a `Tid` and a `Price`.

11.1.1. Parameter

merchantID	90000

mdxi	`<Order>`
	`  <Tid>cust9126</Tid>`
	`  <Price>10.00</Price>`
	`</Order>`

11.1.2. Request

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:etp="https://www.mpay24.com/soap/etp/1.5/ETP.wsdl">
  <soapenv:Header/>
  <soapenv:Body>
    <etp:SelectPayment>
      <merchantID>90000</merchantID>
      <mdxi>
        &lt;Order&gt;
        &lt;Tid&gt;cust9126&lt;/Tid&gt;
        &lt;Price&gt;10.00&lt;/Price&gt;
        &lt;/Order&gt;
      </mdxi>
    </etp:SelectPayment>
  </soapenv:Body>
</soapenv:Envelope>
```

11.1.3. Response

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:etp="https://www.mpay24.com/soap/etp/1.5/ETP.wsdl">
  <SOAP-ENV:Body>
    <etp:SelectPaymentResponse>
      <status>OK</status>
      <returnCode>REDIRECT</returnCode>
      <location>
        https://test.mpay24.com/app/bin/checkout/payment/
        9350ed982ad0050af2353c69a091638a
      </location>
    </etp:SelectPaymentResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

11.2. SelectPayment with long MDXI

In this example, a payment is initiated with a more complex MDXI XML as described in the chapter 6 MDXI XML. Amongst others, it contains a `Tid`, a `Price`, a `TemplateSet`, a `ShoppingCart`, a `BillingAddr` and an `URL` list.

11.2.1. Parameter

merchantID	90000

mdxi	`<Order>`
	`  <UserField>User Field 2012-09-04 09:33:39</UserField>`
	`  <Tid>cust0172</Tid>`
	`  <TemplateSet Language="EN">WEB</TemplateSet>`
	`  <ShoppingCart>`
	`    <Description>Test shopping cart</Description>`
	`    <Item>`
	`      <ProductNr>001</ProductNr>`
	`      <Description>Test product A</Description>`
	`      <Package>Box A</Package>`
	`      <Quantity>1</Quantity>`
	`      <ItemPrice Tax="0.20">15.00</ItemPrice>`
	`      <Price>15.00</Price>`
	`    </Item>`
	`    <ShippingCosts>7.50</ShippingCosts>`
	`    <Tax>3.00</Tax>`
	`  </ShoppingCart>`
	`  <Price>22.50</Price>`
	`  <Currency>EUR</Currency>`
	`  <BillingAddr Mode="ReadOnly">`
	`    <Name>John Doe</Name>`
	`    <Street>Billing Street 1</Street>`
	`    <Zip>1234</Zip>`
	`    <City>Vienna</City>`
	`    <Country Code="AT"/>`
	`    <Email>billing@test.com</Email>`
	`  </BillingAddr>`
	`  <URL>`
	`    <Success>https://www.hotelmuster.com/success.php`
	`      </Success>`
	`    <Error>https://www.hotelmuster.com/error.php/</Error>`
	`    <Confirmation>https://www.hotelmuster.com/confirm.php`
	`      </Confirmation>`
	`  </URL>`
	`</Order>`

11.2.2. Request

Parts of the MDXI XML have been omitted.

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:etp="https://www.mpay24.com/soap/etp/1.5/ETP.wsdl">
  <soapenv:Header/>
  <soapenv:Body>
    <etp:SelectPayment>
      <merchantID>90000</merchantID>
      <mdxi>
        &lt;Order&gt;
        &lt;UserField&gt;User Field 2012-09-04 09:33:39&lt;/UserField&gt;
        &lt;Tid&gt;cust0172&lt;/Tid&gt;
        &lt;TemplateSet Language=&quot;EN&quot;&gt;WEB&lt;/TemplateSet&gt;
        &lt;ShoppingCart&gt;
        ...
        &lt;/ShoppingCart&gt;
        &lt;Price&gt;22.50&lt;/Price&gt;
        &lt;Currency&gt;EUR&lt;/Currency&gt;
        &lt;BillingAddr Mode=&quot;ReadOnly&quot;&gt;
        ...
        &lt;/BillingAddr&gt;
        &lt;URL&gt;
        &lt;Success&gt;https://www.hotelmuster.com/.../success.php&lt;/Success&gt;
        &lt;Error&gt;https://www.hotelmuster.com/.../error.php&lt;/Error&gt;
        &lt;Confirmation&gt;https://www.hotelmuster.com/.../confirm.php
        &lt;/Confirmation&gt;
        &lt;/URL&gt;
        &lt;/Order&gt;
      </mdxi>
    </etp:SelectPayment>
  </soapenv:Body>
</soapenv:Envelope>
```

11.2.3. Response

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:etp="https://www.mpay24.com/soap/etp/1.5/ETP.wsdl">
  <SOAP-ENV:Body>
    <etp:SelectPaymentResponse>
      <status>OK</status>
      <returnCode>REDIRECT</returnCode>
      <location>
        https://test.mpay24.com/app/bin/checkout/payment/
        cc7fac4a1d7f191775874e209504bf2c
      </location>
    </etp:SelectPaymentResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

11.3. Styling of SelectPayment with MDXI

In this example, a payment is initiated with a comprehensive MDXI XML as described in the chapter 6 MDXI XML. Amongst others, it contains detailed descriptions about how the generated mPAY24 PayPage (return location) should look like.

Please feel free to use or modify the described styling example.

11.3.1. MDXI

```
<Order
  LogoStyle="border: none;"
  PageHeaderStyle="background-color: #FFFFFF; text-align: center; padding-top:20px;
    border-top-left-radius: 5px; border-top-right-radius:5px; font-size: 12px;"
  PageCaptionStyle="display:none;"
  PageStyle="border:none;background-color:#FFFFFF;height: 200px;
    border-bottom-right-radius: 5px; border-bottom-left-radius: 5px;"
  InputFieldsStyle="border:1px solid #BFBFBF; padding:2px 2px 1px 4px; margin:2px"
  DropDownListsStyle="background-color: #FFFFFF; border: 1px solid #BFBFBF;
    height: 21px; margin: 2px"
  FooterStyle="background-color: none; margin-top:15px"
  ButtonsStyle="background-color:#336699; border:1px solid #336699; color:#FFFFFF;
    border-radius: 5px; margin: 20px; font-size: 13.5px; cursor: pointer;"
  Style="margin-left: auto; margin-right:auto; width:94%; max-width:988px;
    background-color: #cccccc;">
  <Tid>cust7251</Tid>
  <TemplateSet Language="DE" CSSName="WEB"/>
    <ShoppingCart
    Style="border:none; background-color:#336699; border-bottom-radius:5px;
       font-size: 12px;"
    HeaderStyle="background-color: #336699; border-top-left-radius: 5px;
      border-top-right-radius: 5px;"
    CaptionStyle="display:none;"
    ProductNrStyle="border:none; background-color:#336699; color:#fff;
      padding: 0px 0px 8px 0px;"
    DescriptionStyle="border:none;background-color:#336699; color:#fff;
      padding: 0px 0px 8px 0px;text-align:left;"
    PackageStyle="border:none; background-color:#336699; color:#fff;
      padding: 0px 0px 8px 0px;"
    QuantityStyle="border:none; background-color:#336699; color:#fff;
      padding: 0px 0px 8px 0px;"
    ItemPriceStyle="border:none; background-color:#336699; color:#fff;
      padding: 0px 0px 8px 0px;text-align:left;"
    PriceStyle="border:none; background-color:#336699;color:#fff;text-align:right;
      padding: 0px 20px 8px 0px;">
    <Item>
      <ProductNr Style="border:none; padding-top: 8px; background-color:#fff;">
```

```
        0125</ProductNr>
      <Description Style="border:none; padding-top: 8px; background-color:#fff;
        text-align:left;">Buch</Description>
      <Quantity Style="border:none; padding-top: 8px; background-color:#fff;">
        1</Quantity>
      <ItemPrice Style="border:none; padding-top: 8px; background-color:#fff;
        text-align:left;">3.90</ItemPrice>
      <Price Style="border:none; padding-top: 8px; background-color:#fff;
        padding-right:20px;">3.90</Price>
    </Item>
    <Item>
      <ProductNr Style="border:none; background-color:#fff;">0107</ProductNr>
      <Description Style="border:none; background-color:#fff;">Heft</Description>
      <Quantity Style="border:none; background-color:#fff;">2</Quantity>
      <ItemPrice Style="border:none; background-color:#fff;text-align:left;">
        10.50</ItemPrice>
      <Price Style="border:none; background-color:#fff; padding-right:20px;">
        21.00</Price>
    </Item>
  <Tax
    HeaderStyle="background-color:#fff;padding:10px 3px 10px 0px;text-align:right;"
    Style="border:none; background-color:#fff; padding:10px 20px 10px 0px;">
    6.80</Tax>
  </ShoppingCart>
  <Price
    HeaderStyle="background-color:#fff; width: 100px; padding-bottom:20px;
      font-size: 110%; text-align: right; border-bottom-left-radius: 5px;"
    Style="border:none; background-color:#fff; border-bottom-right-radius: 5px;
      padding-right: 20px; padding-bottom:20px; font-size: 110%;">40.80</Price>
  <Currency>EUR</Currency>
  <Customer Id="10039853"/>
  <BillingAddr Mode="ReadOnly">
    <Name>Max Mustermann</Name>
    <Street>Hauptstraße 123</Street>
    <Zip>1010</Zip>
    <City>Wien</City>
    <Country Code="AT">Österreich</Country>
  </BillingAddr>
  <URL>
    <Success>http://hotelmuster.com/success</Success>
    <Error>http://hotelmuster.com/error</Error>
    <Confirmation>http://hotelmuster.com/confirm</Confirmation>
    <Cancel>http://hotelmuster.com/cancel</Cancel>
  </URL>
</Order>
```

11.3.2. Request

Parts of the MDXI XML have been omitted.

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
```

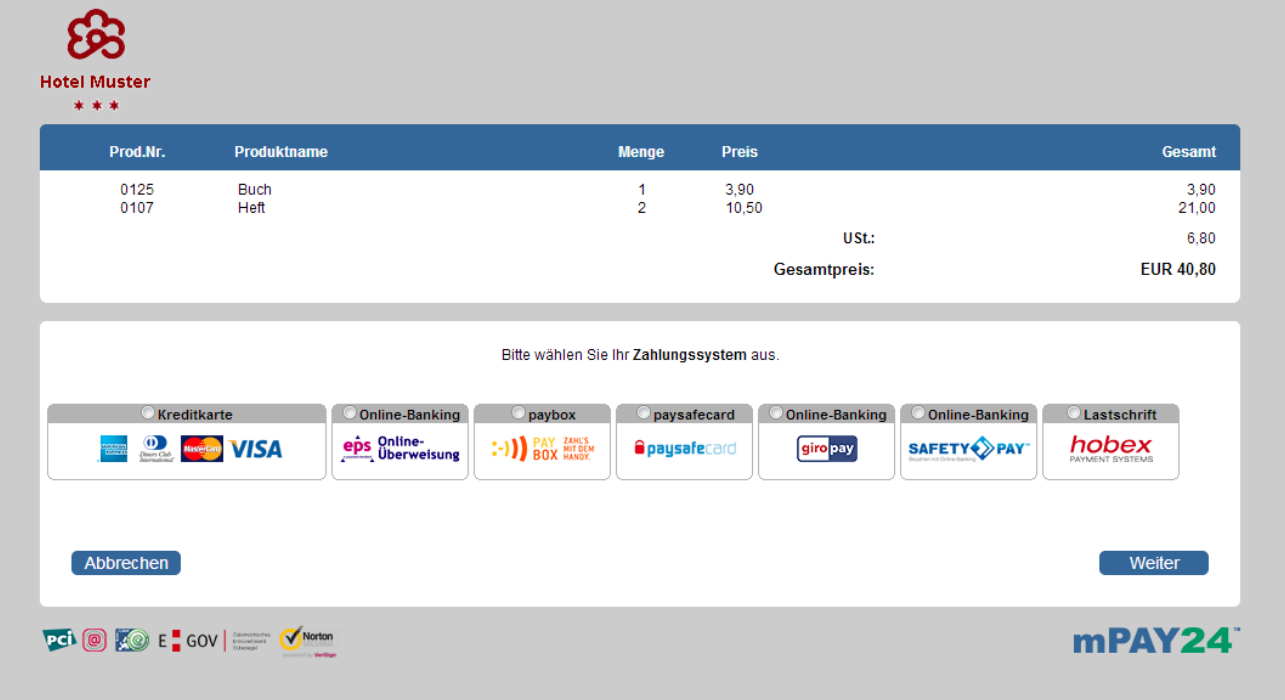
```
xmlns:etp="https://www.mpay24.com/soap/etp/1.5/ETP.wsdl">
<soapenv:Header/>
<soapenv:Body>
  <etp:SelectPayment>
    <merchantID>90000</merchantID>
    <mdxi>
    &lt;Order
      LogoStyle=&quot;border: none;&quot;
      PageHeaderStyle=&quot;background-color: #FFFFFF; text-align: center;
        padding-top:20px; border-top-radius: 5px; font-size: 12px;&quot;
      PageCaptionStyle=&quot;display:none;&quot;
      PageStyle=&quot;border:none;background-color:#FFFFFF; height: 200px;
        border-bottom-right-radius: 5px; border-bottom-left-radius: 5px;&quot;
      InputFieldsStyle=&quot;border:1px solid #BFBFBF; padding:2px 2px 1px 4px;
        margin: 2px&quot;
      DropDownListsStyle=&quot;background-color: #FFFFFF; height: 21px;
        border:1px solid #BFBFBF; margin: 2px&quot;
      FooterStyle=&quot;background-color: none; margin-top:15px&quot;
      ButtonsStyle=&quot;background-color: #336699; border: 1px solid #336699;
        color: #FFFFFF; border-radius: 5px; margin: 20px; font-size:13.5px;
        cursor: pointer;&quot;
      Style=&quot;margin-left: auto; margin-right: auto; width: 94%;
        max-width:988px;  background-color:#cccccc;&quot;&gt;
      &lt;Tid&gt;cust7251&lt;/Tid&gt;
      &lt;TemplateSet Language=&quot;DE&quot; CSSName=&quot;WEB&quot;/&gt;
      &lt;ShoppingCart ... &gt;
        &lt;Item&gt;
          &lt;ProductNr Style=&quot;border:none; padding-top: 8px;
            background-color:#fff;&quot;
          &gt;0125&lt;/ProductNr&gt;
          &lt;Description Style=&quot;background-color:#fff;text-align:left;
            padding-top:8px;border:none;&quot;&gt;Buch&lt;/Description&gt;
          &lt;Quantity Style=&quot;border:none; padding-top: 8px;
            background-color:#fff;&quot;&gt;1&lt;/Quantity&gt;
          &lt;ItemPrice Style=&quot;border:none; background-color:#fff;
            padding-top: 8px;text-align:left;&quot;&gt;3.90&lt;/ItemPrice&gt;
          &lt;Price Style=&quot; background-color:#fff; padding-right:20px;
            border:none; padding-top:8px;&quot;&gt;3.90&lt;/Price&gt;
        &lt;/Item&gt;
        &lt;Item&gt;
          ...
        &lt;/Item&gt;
        &lt;Tax ... &gt;6.80&lt;/Tax&gt;
      &lt;/ShoppingCart&gt;
      &lt;Price ... &gt;40.80&lt;/Price&gt;
      &lt;Currency&gt;EUR&lt;/Currency&gt;
      &lt;Customer Id=&quot;10039853&quot;/&gt;
      &lt;BillingAddr Mode=&quot;ReadOnly&quot;&gt;
        ...
      &lt;/BillingAddr&gt;
      &lt;URL&gt;
        ...
        &lt;/URL&gt;
      &lt;/Order&gt;
      </mdxi>
    </etp:SelectPayment>
  </soapenv:Body>
</soapenv:Envelope>
```

11.3.3. Response

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:etp="https://www.mpay24.com/soap/etp/1.5/ETP.wsdl">
  <SOAP-ENV:Body>
    <etp:SelectPaymentResponse>
      <status>OK</status>
      <returnCode>REDIRECT</returnCode>
      <location>https://www.mpay24.com/app/bin/checkout/payment/
        ef3bbe940e39445fd8f8275f64480da7</location>
    </etp:SelectPaymentResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

11.3.4. mPAY24 PayPage

After redirecting the customer's browser to the location returned by the response, the following mPAY24 PayPage is shown.



Hotel Muster

Prod.Nr.	Produktname	Menge	Preis		Gesamt
0125	Buch	1	3,90		3,90
0107	Heft	2	10,50		21,00
				USt.:	6,80
				Gesamtpreis:	EUR 40,80

Bitte wählen Sie Ihr **Zahlungssystem** aus.

○ Kreditkarte ○ Online-Banking ○ paybox ○ paysafecard ○ Online-Banking ○ Online-Banking ○ Lastschrift

Figure 11: Blue styled mPAY24 PayPage

Appendix

Contact

For complaints, suggestions, data enquiries and assistance:

General enquiries

Phone:	+43 (0)1/513 66 33 -100	**Address:**
Fax:	+43 (0)1/513 66 33 -200	mPAY24 GmbH
E-Mail:	office@mpay24.com	Annagasse 5
Web:	`https://www.mpay24.com`	A-1010 Wien

Marketing & Sales

Phone:	+43 (0)1/513 66 33 -600
E-Mail:	info@mpay24.com
Web:	`https://www.mpay24.com`

Regarding this document and technical assistance:

Phone:	+43 (0)1/513 66 33 -669
Fax:	+43 (0)1/513 66 33 -200
E-Mail:	support@mpay24.com

The contact details stated above will only reply within the mPAY24 office hours:

- Sales: Monday till Friday (working days) 9am till 6pm CET/CEST
- Support: Monday till Friday (working days) 9am till 5pm CET/CEST

Please feel free to send an E-Mail regarding your question any time.

Contact details for the mPAY24 24/7 support team are issued upon an optional legal agreement.

Applicable time zone

All times within this document refer to CET/CEST.

Consideration

Sensitive data and passwords

Please keep all your personal and sensitive data secure. mPAY24 never requests any sensitive payment data or passwords from you. mPAY24 recommends the use of PGP (*Pretty Good Privacy*) or compatible software like GPG (e.g. *GnuPG*) for communication via E-Mail.

Please contact the mPAY24 support about its PGP public key.

URLs

Since the URLs specified below are requested by the customer and therefore must be considered as highly unreliable and insecure. Please use the confirmation interface (refer to 4 Confirmation Interface) instead to determine the state of a transaction.

The URLs are supplied to mPAY24 upon request. Any parameter may be appended. mPAY24 specific parameters are appended after the parameters defined by the merchant. New parameters may be added without notice.

The customer's browser requests the URLs via HTTP GET.

URL	Description
SuccessURL	The following parameters are appended by mPAY24 after parameters defined by the merchant (if any): TID, LANGUAGE, USER_FIELD, BRAND
ErrorURL	The following parameters are appended by mPAY24 after parameters defined by the merchant (if any): TID, LANGUAGE, USER_FIELD, BRAND, ERRTEXT, EXTERNALSTATUS
CancelURL	No special parameters added by mPAY24

Disclaimer

mPAY24 has endeavoured all reasonable measures to assure that all supplied information within this specification is complete and correct at the time of compilation. Unintended and accidental errors may occur, for which we apologize.

mPAY24 cannot be held liable for and does not guarantee for the actuality, correctness and/or completeness of the information provided within this document.

mPAY24 reserves the right to alter, remove and/or add information within this document without further notice. Please contact the mPAY24 support to obtain the latest copy of this document.

Distribution level of this documentation

The following rules and regulations apply to the handling of this document:

Distribution level: **restricted/confidential**

This document may be distributed only to mPAY24 customers (*„merchants“*), partners and technical personnel which is assigned with the implementation of mPAY24.

Any further distribution of this document requires written acknowledgement by mPAY24. Please do not hesitate to contact the mPAY24 support regarding any questions concerning this document.

Supported languages

ISO Code	Language	mPAY24 MI/F	mPAY24 PayPage
BG	Bulgarian		X
CS	Czech		X
DA	Danish		X

DE	German	X	X
FI	Finnish		X
EN	English	X	X
ES	Spanish		X
EL	Greek		X
FR	French		X
HR	Croatian		X
HU	Hungarian		X
IT	Italian		X
JA	Japanese		X
NL	Dutch		X
NO	Norwegian		X
PL	Polish		X
PT	Portuguese		X
RO	Romanian		X
RU	Russian		X
SK	Slovak		X
SL	Slovenian		X
SR	Serbian		X
SV	Swedish		X
TR	Turkish		X
UK	Ukrainian		X
ZH	Chinese		X

The fallback for not supported languages is English.

Supported currencies

All currency codes as proposed by ISO 4217 : `http://www.iso.org/iso/home/standards/currency_codes.htm` are supported by mPAY24.

Using any currency for the transaction then defined in the agreement between the merchant and the acquirer can result in unexpected transaction fees for both, the credit card holder and the merchant.