PAYONE
more than payment.

SPECIFICATIONS SHOP INTEGRATIONS

Integration guide to the PAYONE Platform for shop systems

Version 1.22

About this document

This document describes the integration guidelines applicable to PAYONE integrations in shop systems

Contact

PAYONE GmbH
Fraunhoferstraße 2-4
24118 Kiel, Germany
Tel.: +49 (0) 431 25968-0
Fax: +49 (0) 431 25968-100
Internet: www.payone.de
E-mail: info@payone.de

Brand names

Protection of information

Disclaimer

This document has been prepared with the utmost care. There is, however, no warranty as to its complete accuracy.

Amendments

PAYONE GmbH reserves the right to make product improvements and other changes to the customary extent without notice, provided that they are not unreasonable for the contractual partner.

Liability and License Agreement

The PAYONE payment extension for Magento is provided free of charge by PAYONE GmbH. Please note the PAYONE GmbH liability and license conditions for shop extensions. Should you not accept these terms and conditions, do not install or do not use the extension.

Copyright

Table of contents

1 Changelog

Date	Processor	Description
27.01.2014	Timo Kuchel	SEPA
14.04.2014	Timo Kuchel	Informascore added
11.06.2014	Timo Kuchel	Klarna-Installment added
06.10.2014	Timo Kuchel	• PayPal ECS added • P24 added

17.11.2014	Timo Kuchel	• Globals chenged
		  ○ Dynamic texts
		  ○ Booking text
		  ○ PDF - Download
		• Creditcard validity added

11.12.2014	Timo Kuchel	Yapital added
15.01.2014	Timo Kuchel	PayPal adresstransfer
26.01.2015	Timo Kuchel	Risk configuration changed Mapping countrys
23.02.2015	Timo Kuchel	Changed Yapital. Widget added.
09.03.2015	Timo Kuchel	Creditcard (Channel Frontend) added
17.04.2015	Timo Kuchel	Yapital Widget image updated
28.05.2015	Timo Kuchel	Changed Creditcard Channel Frontend
17.06.2015	Timo Kuchel	Changed Creditcard Channel Client-API
22.07.2015	Florian Bender	Translation of this document, minor amendments and corrections
01.12.2015	Timo Kuchel	• Added Paydirekt completely • Delete Yapital • Added P24, Paydirekt in Payment Methods Overview
16.03.2016	Florian Bender	Added Payolution backend configuration mockups
07.04.2016	Florian Bender	Added Payolution frontend mockups and processes
19.04.2016	Florian Bender	Added section 2.1 Coding standards & secure software

2 Introduction

The aim of this document is to provide a seamless integration with all features and functions available on the PAYONE platform.

Currently there are three ways to communicate with the PAYONE platform:

- Frontend
- Server API
- Client API

For a seamless integration the Server API has to be used in conjunction with the Client API. The description of this APIs is not part of this document. Separate descriptions for the APIs are available.

The Server API is used for most of the transactions, like authorizing/preauthorizing payments, capturing funds, debits and credit assessment. It is a channel of server-to-server communication where the shop software communicates with the PAYONE platform. Due to the fact that all communication gets routed through the shop software before being transmitted to PAYONE this channel is not suitable for credit card data transmission.

The Client API channel is used for communication directly between the customer's browser and the PAYONE platform. It is thus suitable for transmission of data that must not be stored on the merchant's site, for instance credit card data. However, this channel is not suitable for server-to-server communication as needed for capturing or refunding payments.

Both, the Client and Server API are described in separate documents available from PAYONE on request.

2.1 Coding standards & secure software

Payment is vital for all online shops. Thus, regulations of code quality and security have to be followed at all times. Code has to be maintainable by and accessible for all developers. Documentation is essential. All code should be as simple as possible and concise. Per default, all PHP integrations have to adhere to the following regulations:

- PSR-1 for code interoperability.
- PSR-2 for human readability.
- PSR-4 where no other autoloading standard takes precedence.

There may be cases where other standards take precedence over the PSR regulations. In these cases, the preceding standard has to be used after confirmation from PAYONE. For other languages than PHP, comparable standards have to be used after approval from PAYONE, for instance Google's Java Style Guide.

Code quality will be validated using Phpmetrics. Since a lot of the metrics employed are disputed, they will not be used as a "hard" factor for acceptance, but will be taken into account.

For the design and development of secure software, the OWASP Developer Guide must be considered. Additionally, PAYONE will provide a document with mandatory standards for secure software.

2.2 Payment methods

PAYONE works to continuously update and expand the features and available payment methods of the PAYONE platform. PAYONE thinks of payment methods as separated into two layers: the payment method itself and the brand or provider for a flavor of this payment methods. Examples would be credit card as a payment method and Visa as a brand; or online bank transfer as a payment method and SOFORT Überweisung as a brand/provider. In PAYONE's lingo, this would be the "type".

For each layer there are specific settings to be kept in mind to ensure a smooth operation. For instance, the online bank transfer type iDeal is only available in the Netherlands and the shop integrator or merchant has to ensure that it only works with that country.

Currently, the following payment methods and types are supported by PAYONE

- Credit and debit cards
    - Visa
    - MasterCard
    - American Express
    - Carte Bleue
    - Diners Club
    - Discover
    - JCB
    - Maestro

- Online bank transfer
    - SOFORT Überweisung
    - giropay
    - eps
    - PostFinance E-Finance
    - PostFinance Card

  - iDEAL
  - P24

- e-Wallet
  - PayPal
  - PayDirekt

- Account based payment methods
  - SEPA Core Direct Debit
  - Invoice
  - Pre-payment
  - Cash on delivery

- Financing
  - Commerz Finanz
  - Payolution
  - RatePay

- Safe invoice
  - BillSAFE
  - Klarna
  - Payolution
  - RatePay

- Safe Direct Debit
  - Payolution
  - RatePay

- Monthly Invoice
    - Payolution

- Cash payment methods
    - Barzahlen

3 Configuration

The respective configurations have to be available for multi shop solutions, e.g. store views in Magento. It should be possible to configure them individually.

3.1 Configuration Export

To enable our support to troubleshoot merchant's problems efficiently, a configuration export feature has to be implemented.

In addition to the configuration of the PAYONE module it is vital to include other installed modules and their versions as well as the versions of the shop software and the PAYONE module itself.

3.1.1 Example

```
<config>
  <shop>
    <name><![CDATA[Deutschland]]></name>
    <system>
      <name>Shop System</name>
      <version>1.2.3.4</version>
      <edition>CE</edition>
      <modules>
        <Logistics>1.2.3</Logistics>
        <Payone>3.4.5</Payone>
        <SuperAccountingModule>0.1.0</SuperAccountingModule>
      </modules>
    </system>
    <global>
      <mid>12345</mid>
      <aid>67890</aid>
      <portalid>1112333</portalid>
      <request_type>authorization</request_type>
      <status_mapping>
        <vor>
          <map from="appointed"
to="pending_payment|pending_payment"/>
          <map from="paid" to="processing|processing"/>
        </vor>
      </status_mapping>
    </global>
    <clearingtypes>
      <vor>
        <title><![CDATA[PAYONE advance_payment]]></title>
        <id>3</id>
        <mid>14648</mid>
        <aid>23679</aid>
        <portalid>2018344</portalid>
        <active>1</active>
        <countries>DE,AT,CH</countries>
        <authorization>authorization</authorization>
        <mode>test</mode>
```

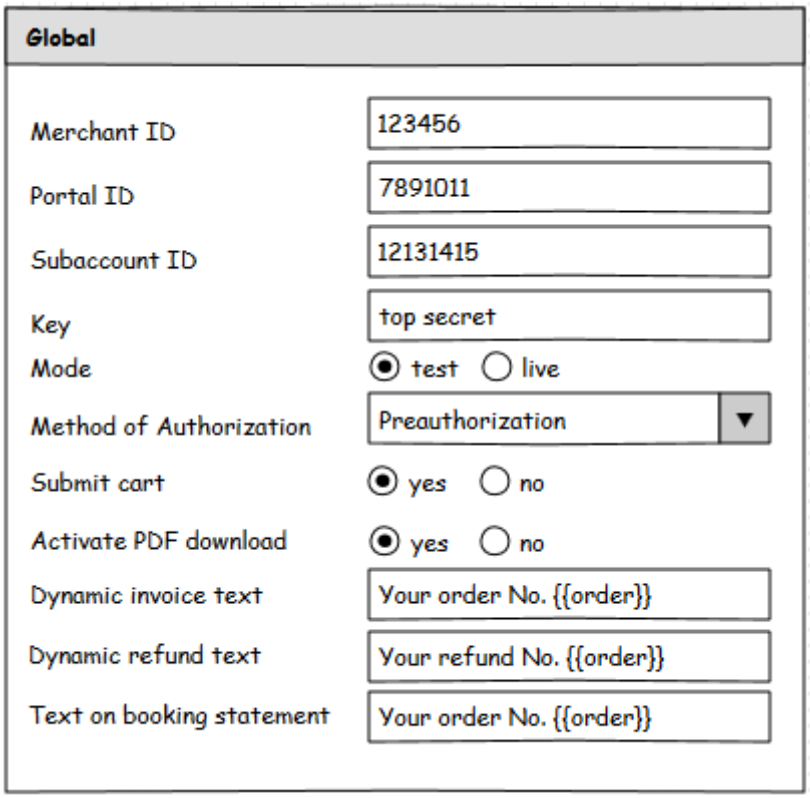
```
      </vor>
    <protect>
      <consumerscore>
        <active>0</active>
        <mode>test</mode>
        <addresscheck></addresscheck>
        <red>advance_payment</red>

<yellow>advance_payment,cash_on_delivery,creditcard,debit_payment,fi
nancing,online_bank_transfer,safe_invoice,wallet</yellow>
      </consumerscore>
      <addresscheck>
        <active>0</active>
        <mode>test</mode>
      </addresscheck>
    </protect>
    <misc>
      <transactionstatus_forwarding>
        <config
status="appointed,capture,paid,underpaid,cancelation,refund,debit,re
minder,vauthorization,vsettlement,transfer,invoice"
url="http://url.weiterleitung.php" timeout="100"/>
      </transactionstatus_forwarding>
    </misc>
  </shop>
</config>
```

3.2 Payment configuration

The following configuration options have to be available globally and for each payment type.

3.2.1 Global Configuration

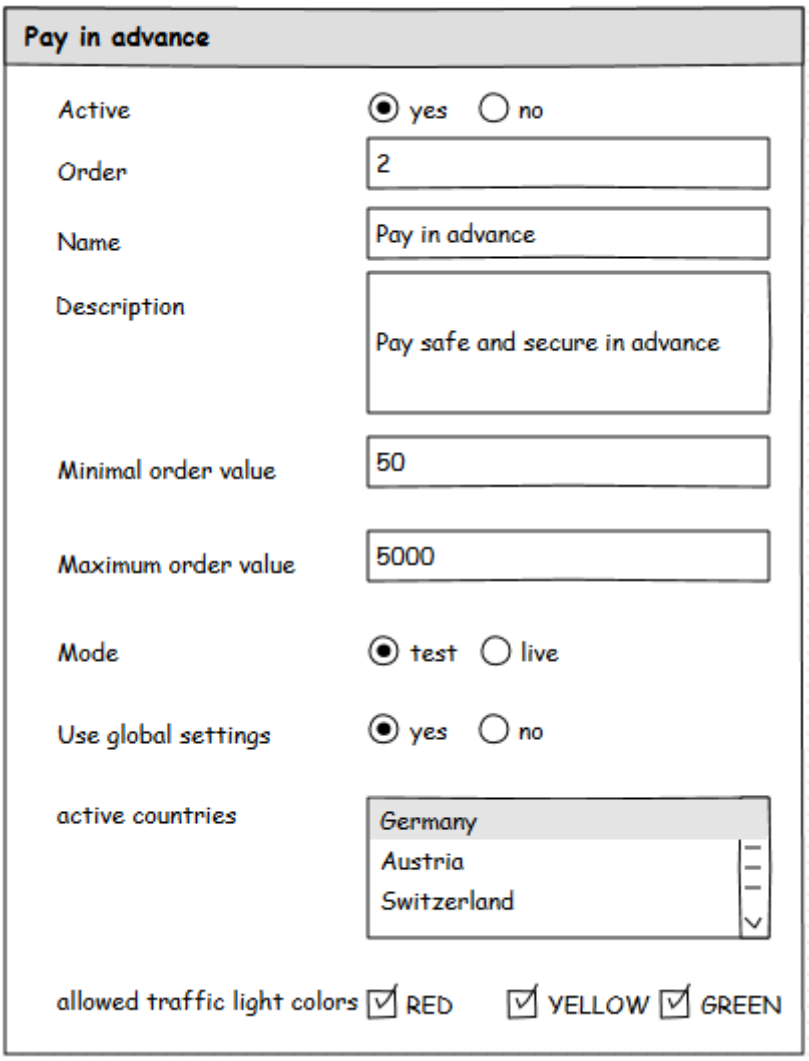


Parameter	Description
Merchant ID	ID of the merchant account
Portal ID	ID of the payment portal
Subaccount ID	ID of the subacction
Key	The key of the payment portal
Mode	• Live = Payment will be processed on the PAYONE Platform • Test = Payment will be simulated on the PAYONE Test Environment and the payment method's sandboxes, if applicable.
Method of authorization	The authorization method must be configureable per payment method. Selection of: • Preauthorization • Authorization The **preauthorization** initiates the transaction on the PAYONE platform. If the payment method allows it, the amount will be reserved (e.g. on the credit card). With payment methods like SOFORT.com the amount will be transferred immediately, as no reservation can be made. For payment methods like invoice or pre-payment the payment process will just be initialized on the PAYONE platform. If **authorization** is selected, the amount will be transferred immediately, if possible.
Submit cart	Selection of: • yes

	• no Determines whether information about the customers articles (name, id, price, VAT, etc.) should be transmitted to PAYONE. Required for some payment methods.
Activate PDF download	If the customer uses the Invoice module of PAYONE, the invoice generated by PAYONE should be available as a download in the backend of the shop. Selection of: • yes • no
Dynamic invoice text	When using the Invoice module of PAYONE, this parameter determines the dynamic part of the invoice. Variables like order number must be replaced with the actual values.
Dynamic refund text	When using the Invoice module of PAYONE, this parameter determines the dynamic part of the debit note. Variables like order number must be replaced with the actual values.
Text on booking statement	This field defines the text that will be visiable on the buyer's account statement. It will be transmitted as the parameter narrative_text in the (pre-)authorization request. Variables like order number must be replaced with the actual values.

The global configuration parameters have to be inherited to the payment method and the individual payment type. Furthermore, it must be possible to copy a payment method to have different configurations for different store views, countries, campaigns, customer groups etc.

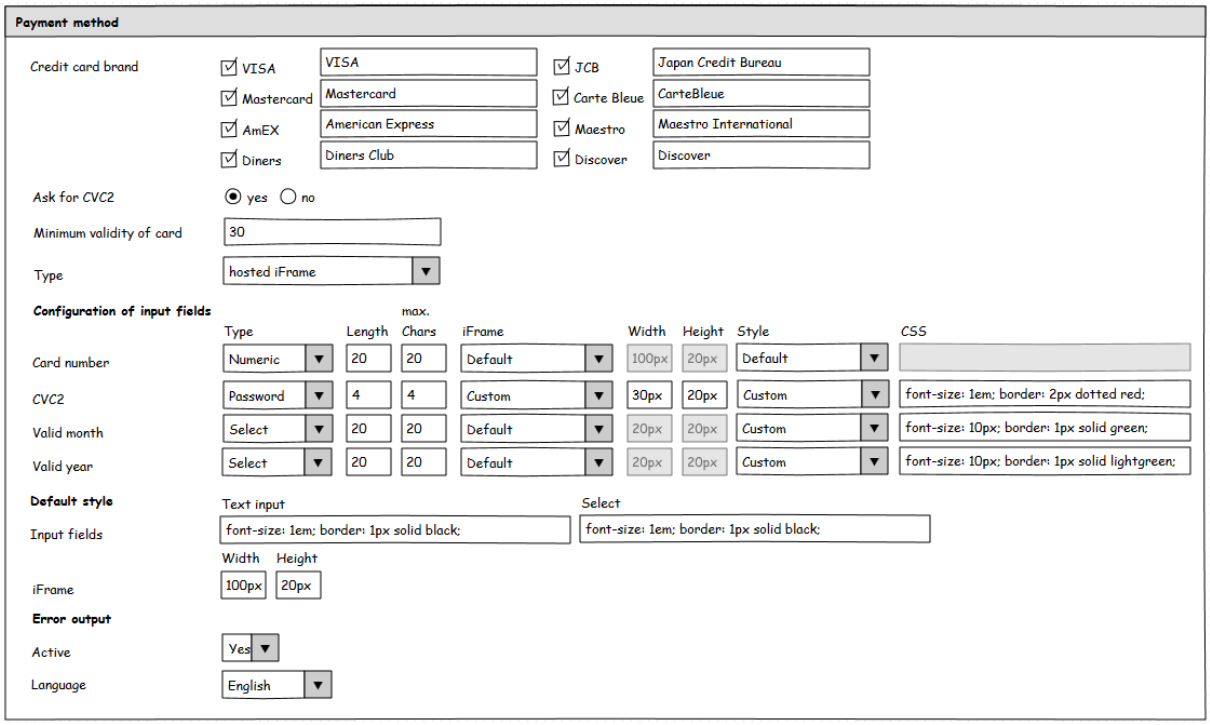
3.2.2 Detail configuration



Parameter	Description
Active	Activates the payment method
Order	Integer that determines the order of the payment methods in checkout
Name	Contains the name of the payment method as it will be displayed in checkout
Description	A description of the payment method, that will be shown to the customer during checkout.
Minimal order value	The minimal order value for this payment method to be available.
Maximum order value	The maximum order value for this payment method to be available.
Mode	• Live = Payment will be processed on the PAYONE Platform • Test = Payment will be simulated on the PAYONE Test Environment and the payment method's sandboxes, if applicable.
Use global settings	Defines whether this payment method uses the global settings (see above). If *no* is selected, the input fields from the global configuration must be displayed.
Active countries	Multiple selection in which countries this payment method should be available.
Allowed traffic light	The traffic light colors for which this payment method is available after a credit rating or

values	address check was performed.

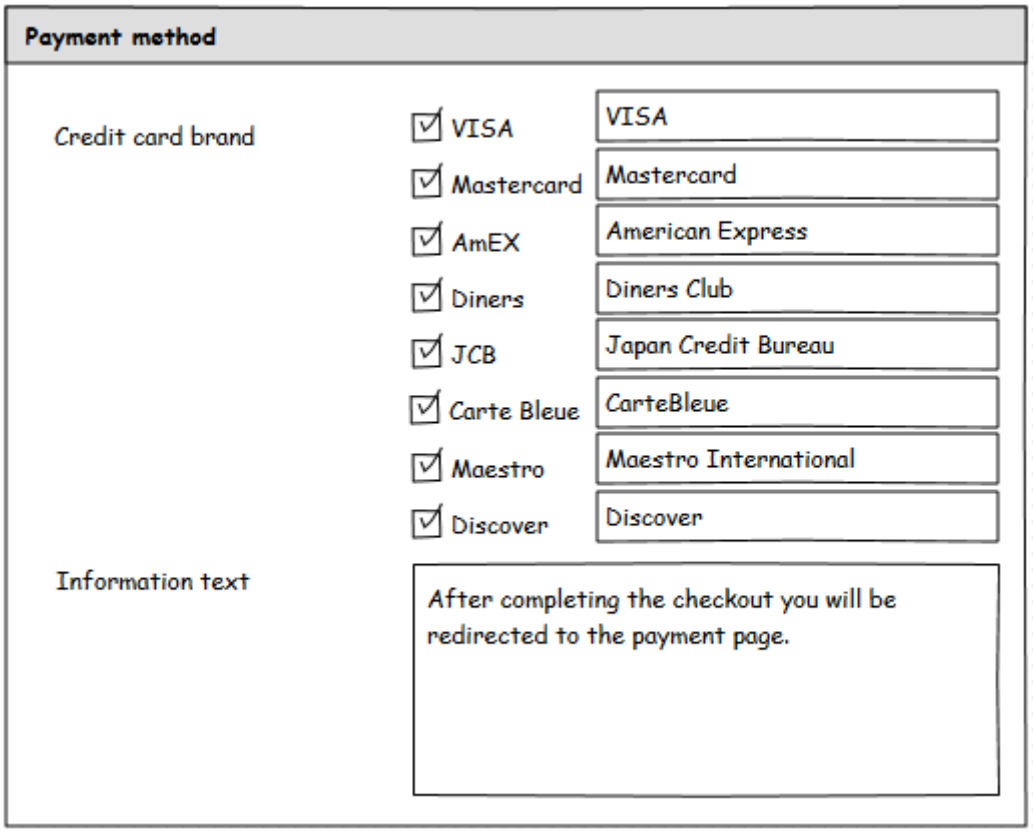
3.2.2.1 Credit card (Client-API Channel)



Parameter	Description
Credit card brand	Activate the credit card brands and allow the merchant to configure lables for the credit card brands
Ask for CVC2	Activate or deactivate the CVC2 field. This usually is governed by the merchant's agreement with the acquirer
Minimum validity of card	Amount of days a credit card has to be valid to be accepted in the shop
Type	Select • hosted iFrame • AJAX
Configuration of input fields	The individual input fields for credit card data can be configured here. The configuration of input fields, default styles, and error output should not be displayed if type "AJAX" is selected.
Type	Select: • Numeric - Only numbers are allowed. For mobile users the number keypad will be used (`input type="tel"`) • Password - `input type="password"` • Text - `input type="text"`
Length	Length of the field in characters (HTML attribute `size`)
Max. Chars	Maximum length of input (HTML attribute `maxlength`)
iFrame	Select: • Default - Uses width and height from default style • Custom - Width and height as defined in the following fields are used
Width	Width in a valid CSS statement.

	Deactived if "Default" is selected.
Height	Height in a valid CSS statement. Deactivated if "Default" is selected.
Style	• Default - CSS from "Default" is used. • Custom - CSS from the following field is used.
CSS	CSS style for input field. Deactivated if "Default" is selected.
Default style	
Input fields	• Input - CSS for input fields (HTML `input`) • Select - CSS for select fields (HTML `select`)
iFrame	• Width - CSS Width • Height - CSS Height
Error output	
Active	Select: • Yes • No If "Yes" is selected, the PAYONE `errormessage` parameter will be shown in a defined container.
Language	Select the language in which the error message should be displayed. If error output is deactivated, this field should not be displayed.

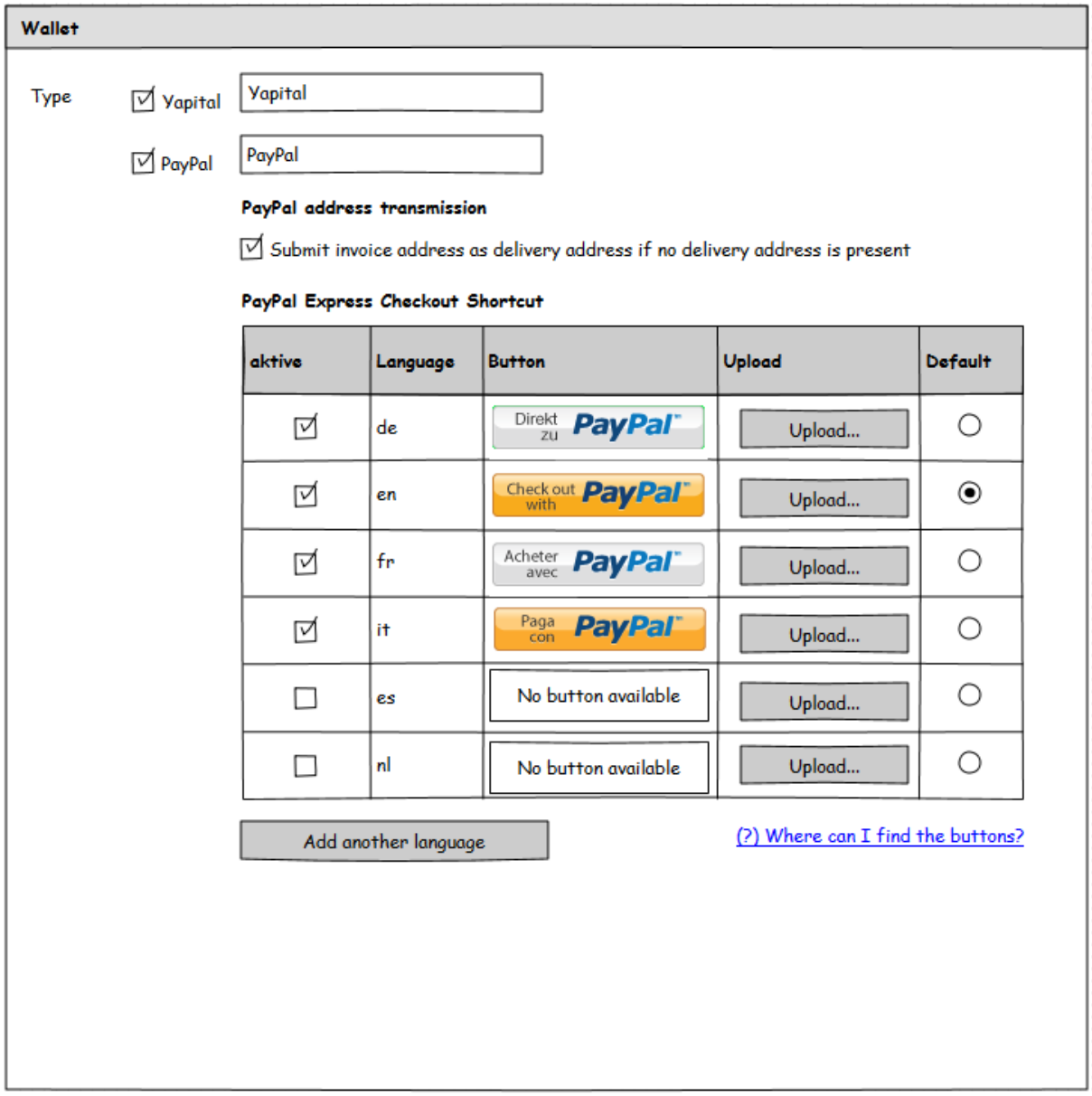
3.2.2.2 Credit card (Channel Frontend)



Parameter	Description
Credit card	Activate the different card brands and assign a name to them displayed to the customer in the

brand	checkout page.
Information text	This text should tell the customer that they will be redirected after confiming the order to enter their card data on a third party site.

3.2.2.3 eWallet

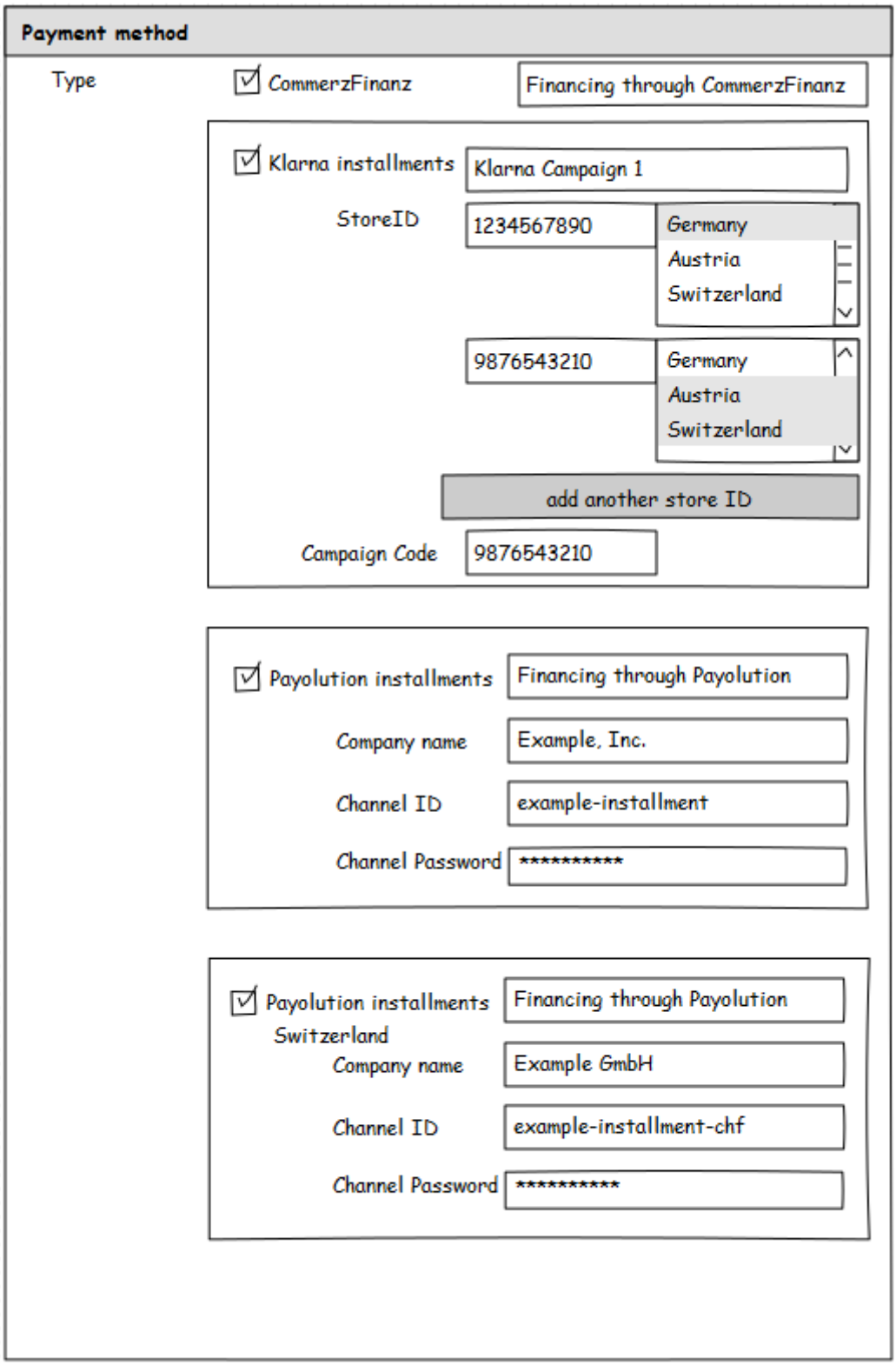


Parameter	Description
Type	Activate the different types of wallets and assign a name to be displayed in checkout.
PayPal address transmission	When this option is activated, the invoice address will be transmitted to PayPal as the delivery address when no other delivery address is specified.
PayPal ECS	

Field	Description
active	activates PayPal ECS
Language	Language of the button
Button	Preview of the button

	Upload	Button to enable the merchant to upload PayPal ECS buttons
	Default	Radio buttons to define the standard button that is used for languages that have no specific button available.
	Add another button	Button to add another row.
	Where can I find the buttons?	Info text describing where to find the PayPal ECS buttons.

3.2.2.4 Financing

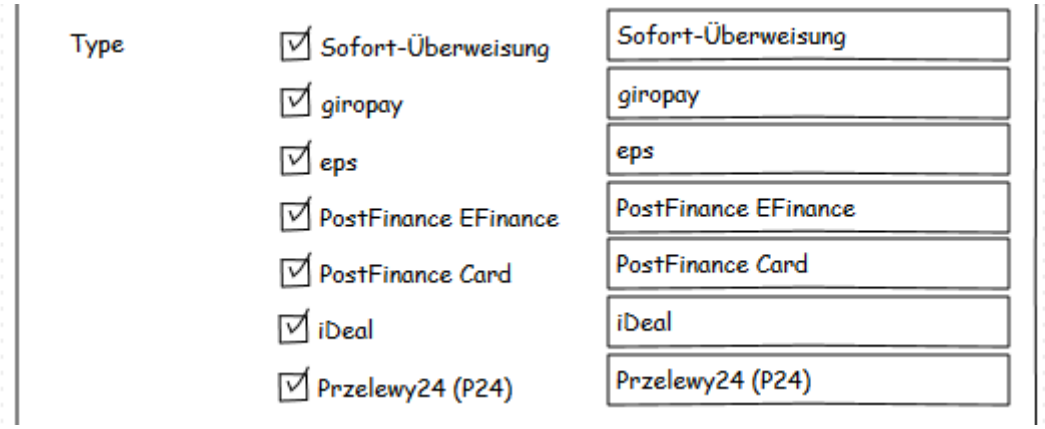


Parameter	Description
Type	Activate the different types of financing and assign a name to be displayed in checkout.
Klarna StoreID	To perform payments through the financing type Klarna part payment, at least one store ID has to be configured for the shop. This ID has to be set up for a specific set of countries as determined by Klarna. The country of the invoice address of the customer determines the country. It must be

	possible to duplicate the payment method to have multiple store IDs for different countries.
Campaign Code	The campaign code determines the interest rate and run time of the part payment and is determined by Klarna.
Company name	The legal full name of the company. It is used to generate the Payolution privacy statement. Should be pre-filled with the company name that is available through the shop system's configuration.
Channel ID	The channel ID for this payment method. Determined by Payolution.
Channel Password	The password for this payment method's channel. Determined by Payolution.

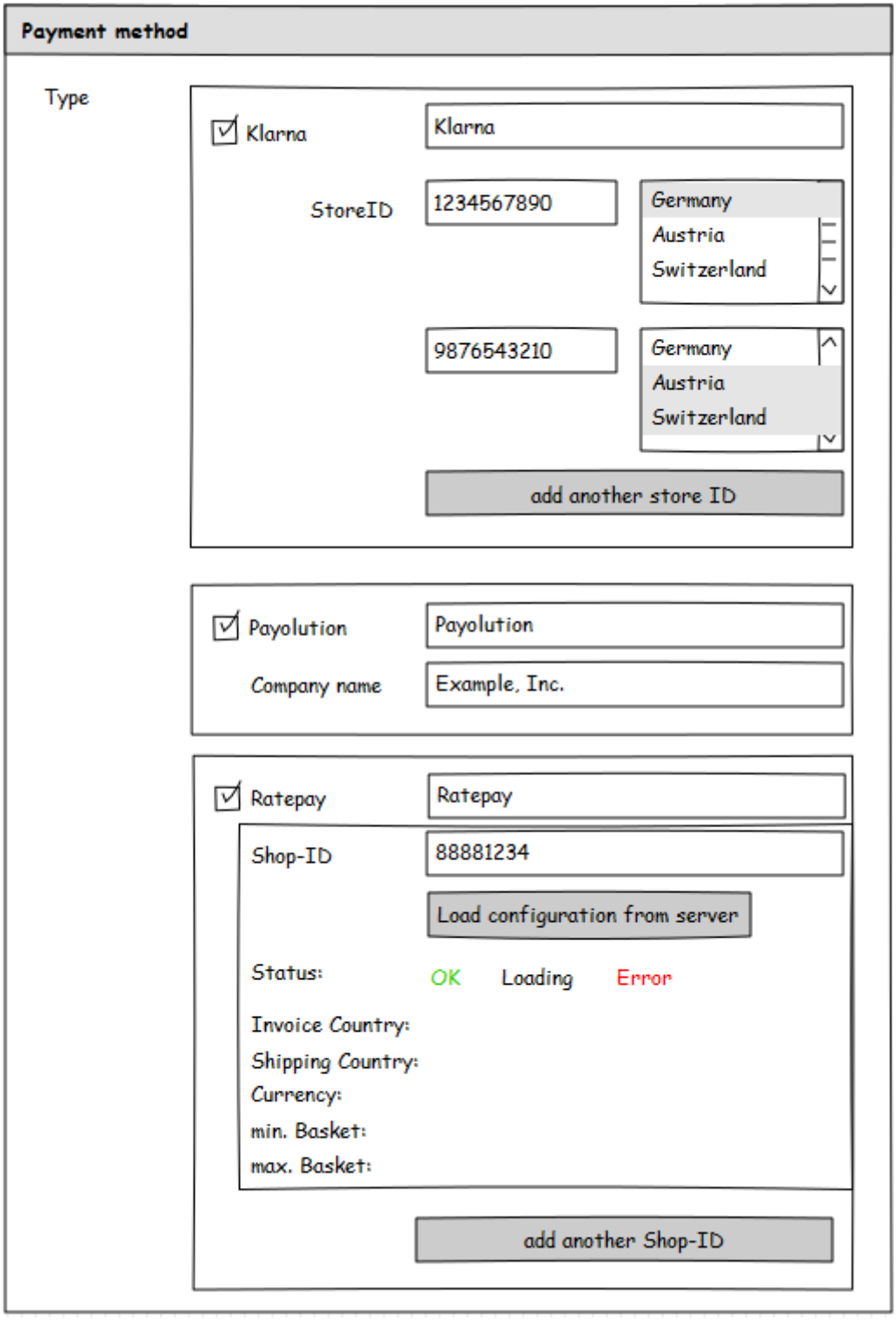
The financing type Klarna part payment must be duplicateable so multiple store IDs and campaign codes can be configured.

3.2.2.5 Online Bank Transfer



Parameter	Desciption
Type	Activate the different types of online bank transfer and assign a name to be displayed in checkout.

3.2.2.6 Safe invoice

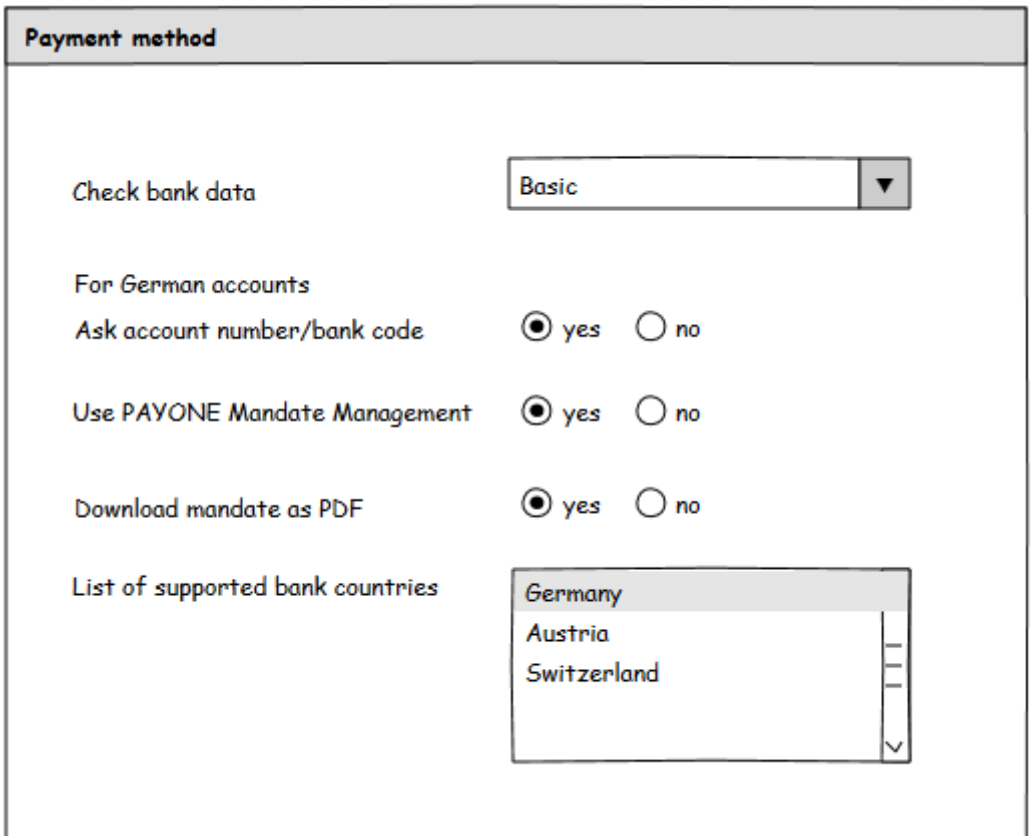


Parameter	Description
Type	Activate the different types of safe invoice and assign a name to be displayed in checkout.
Klarna StoreID	To perform payments through the Klarna safe invoice, at least one store ID has to be configured for the shop. This ID has to be set up for a specific set of countries as determined by Klarna. The country of the invoice address of the customer determines the country. It must be possible to duplicate the payment method to have multiple store IDs for different countries.
Company	The legal full name of the company. It is used to generate the Payolution privacy statement.

name	Should be pre-filled with the company name that is available through the shop system's configuration.
Shop-ID	The shop ID is a numerical identifier that references to a configuration set on RatePay's site. It is used to get country, currency, and basket value information from RatePay's server through the PAYONE platform. The shop ID is determined by RatePay and available through their merchant interface "My RatePay".

After the "Load configuration from server" button has been triggered, a `genericPayment` request with the `add_paydata[action]=profile` will be issued to fetch the data from the server. While that is in progress, the status text has to change to "loading". After the data has successfully been retrieved, the status should be "OK", otherwise it should be "error". The received parameters should be updated in the payment method configuration, e.g. the payment method should only be available in Germany and within the minimum and maximum basket value, etc. It should be possible to define multiple Shop-IDs, but only one Shop-ID can be used in checkout at a time.

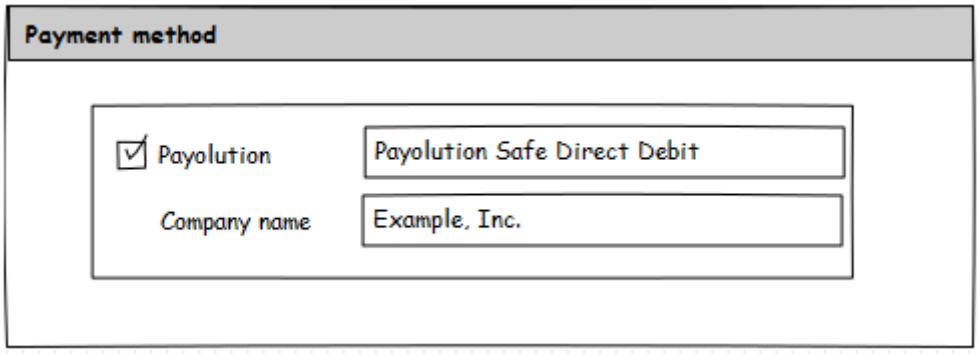
3.2.2.7 SEPA Direct Debit



Parameter	Description
Check bank data	Selection : • Basic • Check POS black list • None (only possible if PAYONE Mandate Management is inactive)
Ask account number/bank code	Optionally, account number and bank code can be entered additionally for IBAN/BIC (German accounts only)
Use PAYONE	Defines, if the shop should check for an existing SEPA mandate using the *managemandate*

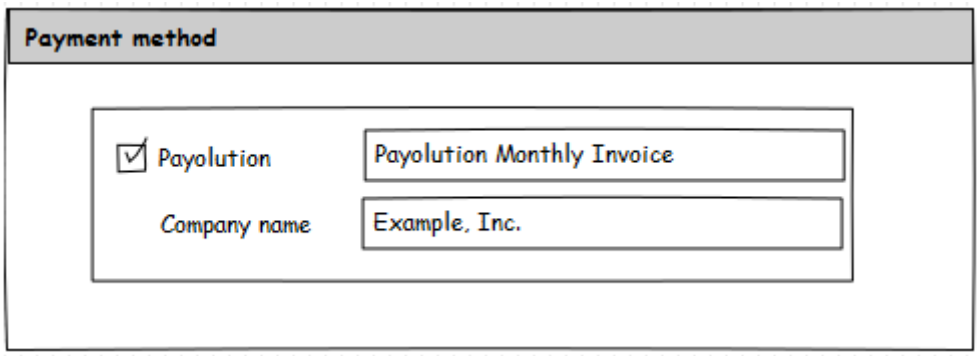
Mandate Management	request or whether a temporary mandate should be created. If a mandate already exists, the parameter mandate_status=active will be returned and the mandate must not be approved again by the account owner.
Download mandate as PDF	Defines if the buyer can download the PDF SEPA mandate after the order is completed. This is only possible if the product "SEPA Mandate as PDF" was purchased from PAYONE. PAYONE checks if a mandate is present for this account or if a new one has to be created. Only then the mandate is newly created. However, in both cases, the mandate can be downloaded using the request *getfile.*
List of supported bank countries	List of allowed bank countries. Multiple select: 33 SEPA countries.

3.2.2.8 Safe Direct Debit



Parameter	Description
Type	Activate the different types of safe direct debit and assign a name to be displayed in checkout.
Company name	The legal full name of the company. It is used to generate the Payolution privacy statement. Should be pre-filled with the company name that is available through the shop system's configuration.

3.2.2.9 Monthly Invoice



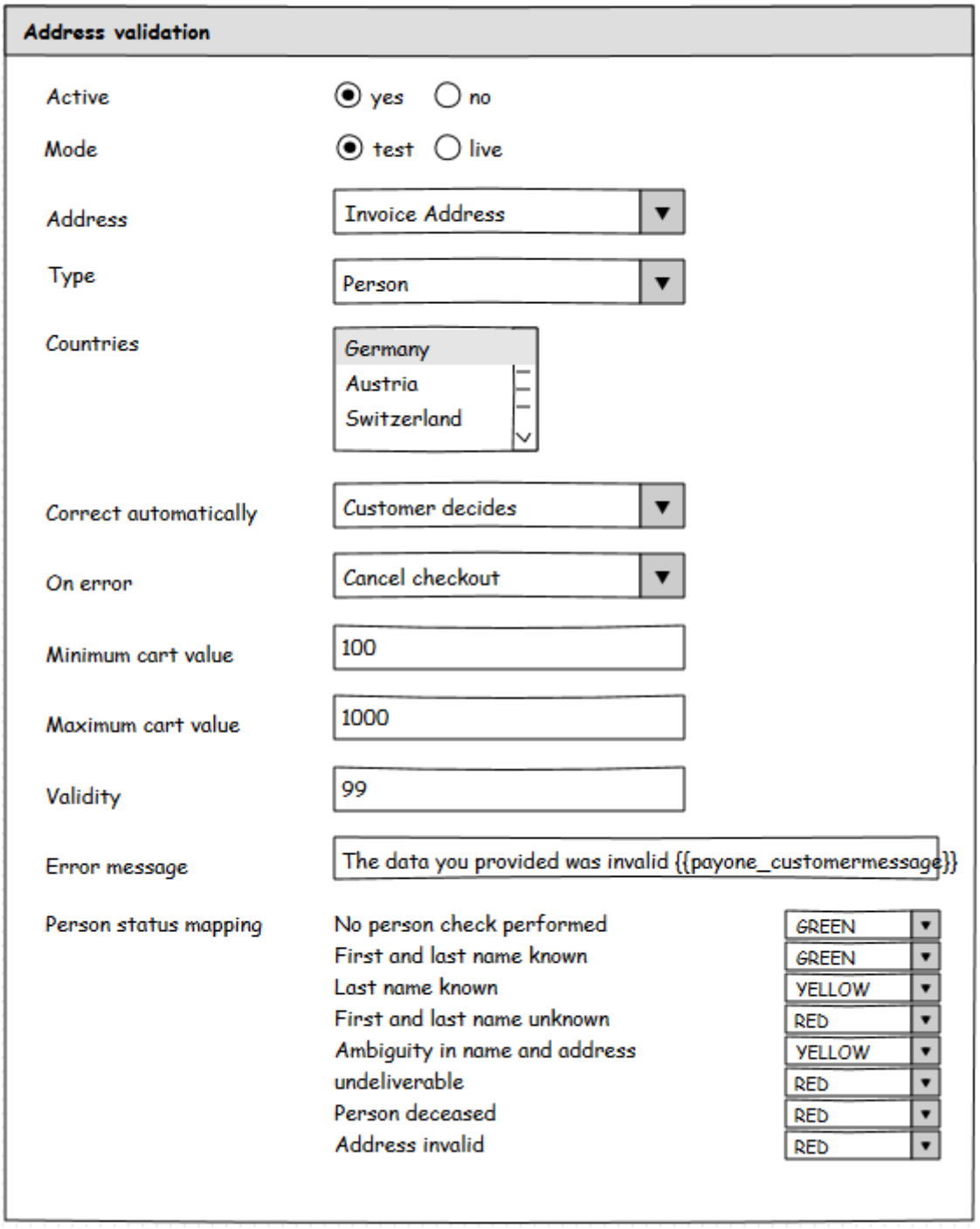
Parameter	Description
Type	Activate the different types of monthly invoice and assign a name to be displayed in checkout.
Company name	The legal full name of the company. It is used to generate the Payolution privacy statement. Should be pre-filled with the company name that is available through the shop system's configuration.

3.3 Configuring the risk checks (Module "Protect")

Configuration sets of address and credit checks must be duplicateable, in order to provide different configuration possibilities for different countries, or different types of checks.

3.3.1 Address checks

Address checks verify addresses that the customer enters during the checkout process. The following options must be available:



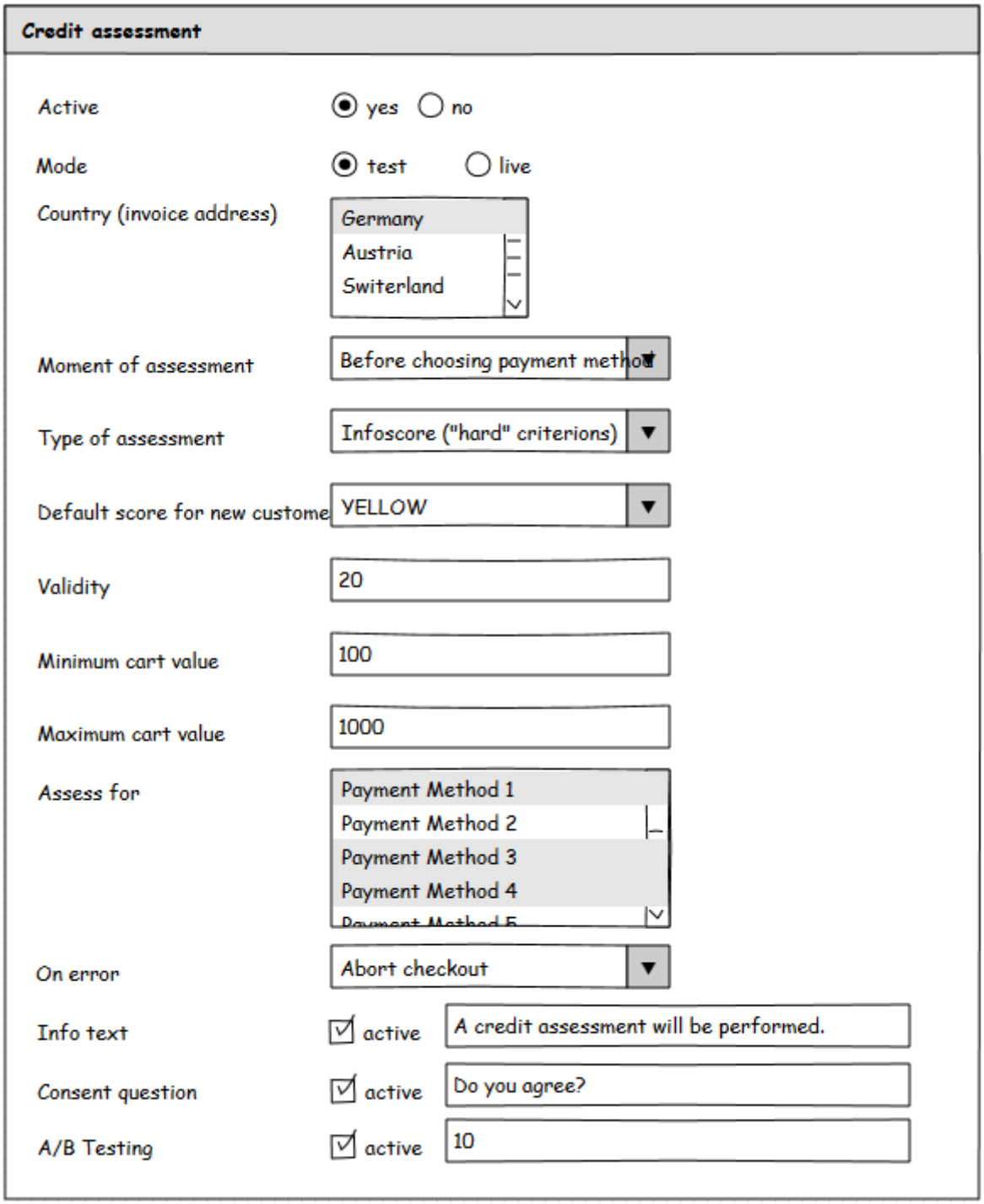
Parameter	Description
Active	Activates address validation in the shop
Mode	• Live = A real check will be performed with the given data • Test = Only a specific set of test data is allowed and will give certain results
Address	For which address is this configuration valid? Selection from

	• Invoice address • Delivery address
Type	Wich type of address check should be performed in this configuration? Selection from: • Basic • Person ( only available in Germany)
Countries	Countries for this configuration set.
Correct automatically	What happens when the PAYONE platform returns the possibility of an address correction (secstatus=20)? Should it be corrected automatically? • Yes • No • Customer decides (in this case the customer can correct their address or approve the correction suggested by the address check system. A following address check should not be performed.)
On error	• Cancel checkout • Request re-entry of the address causing the problem • Perform follow-up credit check • Continue
Minimum cart value	The cart value minimally needed to perform an address check.
Maximum cart value	For cart values higher than this value, no address checks will be performed (for example when a credit check is performed instead).
Validity	The amount of days the result of an address check should be cached for registered customers.
Error message	The info message that will be displayed to customers during checkout if incorrect data was entered. There must be a variable placeholder which will be replaced by the message the PAYONE platform sends.
Person status mapping	Depending on the result of the address check "person", a traffic light value can be assigned to a retain response. Using this the payment method portfolio can be restricted.

3.3.2 Credit check

The credit check determines the creditworthiness of a customer. It can be either performed for high-risk payment methods like invoice, or, in order to create a better checkout experience for customers, just present them with the payment methods suiting their credit rating.

The following configuration options must be available in the backend:



Parameter	Description
Active	Activates address validation in the shop
Mode	• Live = A real check will be performed with the given data • Test = Only a specific set of test data is allowed and will give certain results
Country (Invoice address)	The countries in which this configuration set is applied.
Moment of assessment	• Before choosing payment method • After choosing payment method

Type of assessment	• Infoscore ("hard" criterions) • Infoscore (all features) • Infoscore (all features + Boniscore)
Default score for new customers	The default traffic light score for new customers if no credit assessment has been performed
Minimum cart value	The cart value minimally needed to perform an address check.
Maximum cart value	For cart values higher than this value, no address checks will be performed (for example when a credit check is performed instead).
Validity	The amount of days the result of an address check should be cached for registered customers.
Assess for	Multiple selection for which payment method the credit check should be performed. This selection only is available if "after choosing payment method" is selected as the moment of assessment.
On error	• Abort checkout • Continue
Info text	This text will be displayed before a credit check is performed.
Consent question	Text for a consent question displayed for the customer. The customer has to decide during checkout between the options "Yes" or "No", none of which may be pre-selected. If the customer selects "No", they are assigned the default traffic light value.
A/B Testing	If activated, every nth customer will be tested.

Which payment method is available to which credit rating can be configured for the specific payment methods. This should be defined by a number in the range from 0 to 1000 (0 being the worst credit rating and 1000 the best). If no Boniscore is returned, the traffic light values will be converted to a score value:

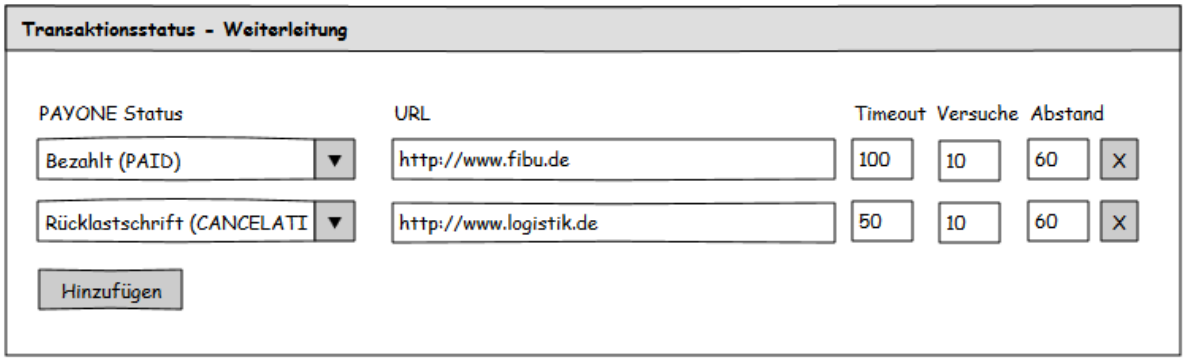
score **traffic light**	**scorevalue** **Boniscore**
GREEN	500
YELLOW	300
RED	100

3.4 Processing the transaction status

The transaction status represents the status of a transaction on the PAYONE platform. To ensure the status notifications are processed the right way, the following configuration options must be available:

3.4.1 Forwarding the transaction status

The transaction status can only be sent to one URL by the PAYONE platform. In most use cases, however, it is required to inform multiple systems about the status of a transaction, for instance an ERP system or logistics software.

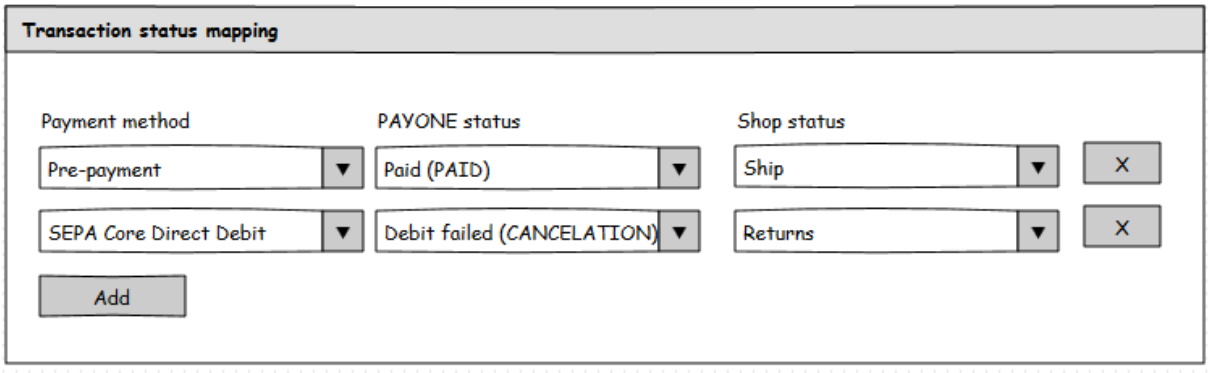


Parameter	Description
PAYONE Status	The status notification PAYONE sends to the shop
URL	The target URL for forwarding the transaction status
Timeout	Timeout in seconds
Amount of tries	The number of times the shop should try to connect to the target system.
Interval	The interval in seconds after that another connection attempt will be made

All unsuccessful forwarding attempts must be logged, either separately or in the transaction dataset. The logging guidelines apply.

3.4.2 Mapping

The mapping configuration of the transaction status maps a PAYONE transaction status to a shop status.



4 Processes

4.1 General information on processes

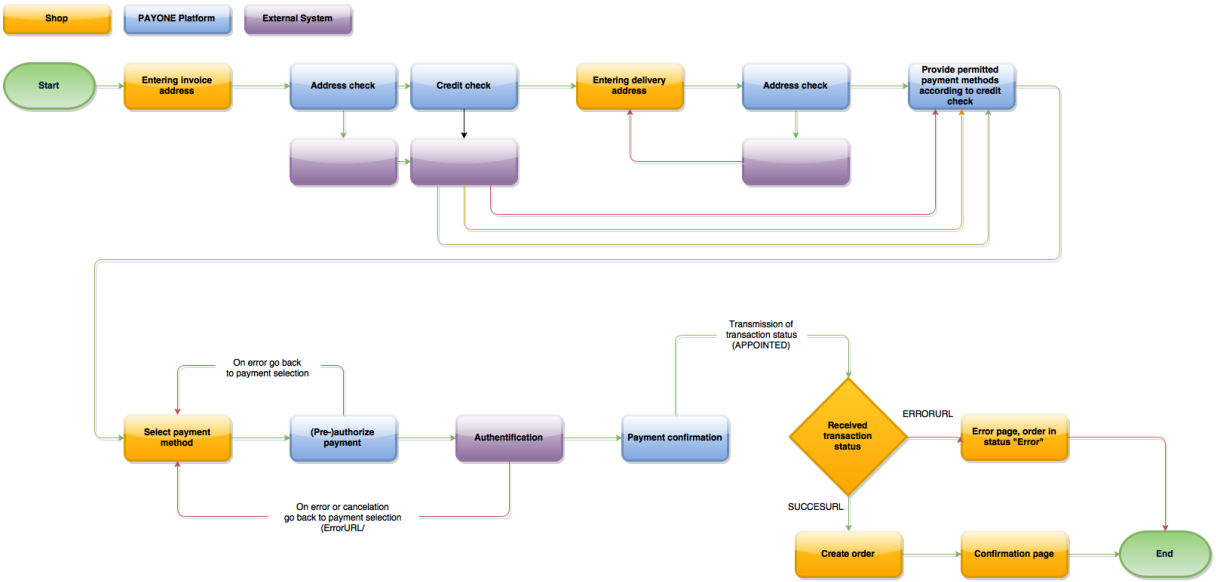
4.1.1 B2C and B2B transactions, VAT

In a normal B2C scenario in European countries, all prices shown in the shop include VAT. Thus, prices and amounts are submitted to PAYONE including VAT, together with the VAT basis points using the va[n] parameter (for example va[1]=1900; cf. API Description for details). However, there might be B2B shop cases where prices in the shop frontend do not include VAT (as in some jurisdictions, for instance Germany, there is no VAT for companies), but still prices submitted to PAYONE have to include VAT as companies pay the full amount first and then get back the overpaid VAT from the tax authorities. However, this does not apply if a VAT ID is submitted. The following three cases have to be accounted for:

- Standard B2C: All prices and amounts submitted to PAYONE include VAT, prices displayed in shop include VAT, va[n] parameter is set to the VAT basis points.
- Standard B2B: All prices and amounts submitted to PAYONE include VAT, prices displayed in shop may or may not include VAT, va[n] parameter is set to the VAT basis points.
- B2B w/VAT ID: All prices and amounts submitted to PAYONE do not include VAT, prices displayed in shop may or may not include VAT, va[n] parameter is set to 0, vatid parameter is mandatory.

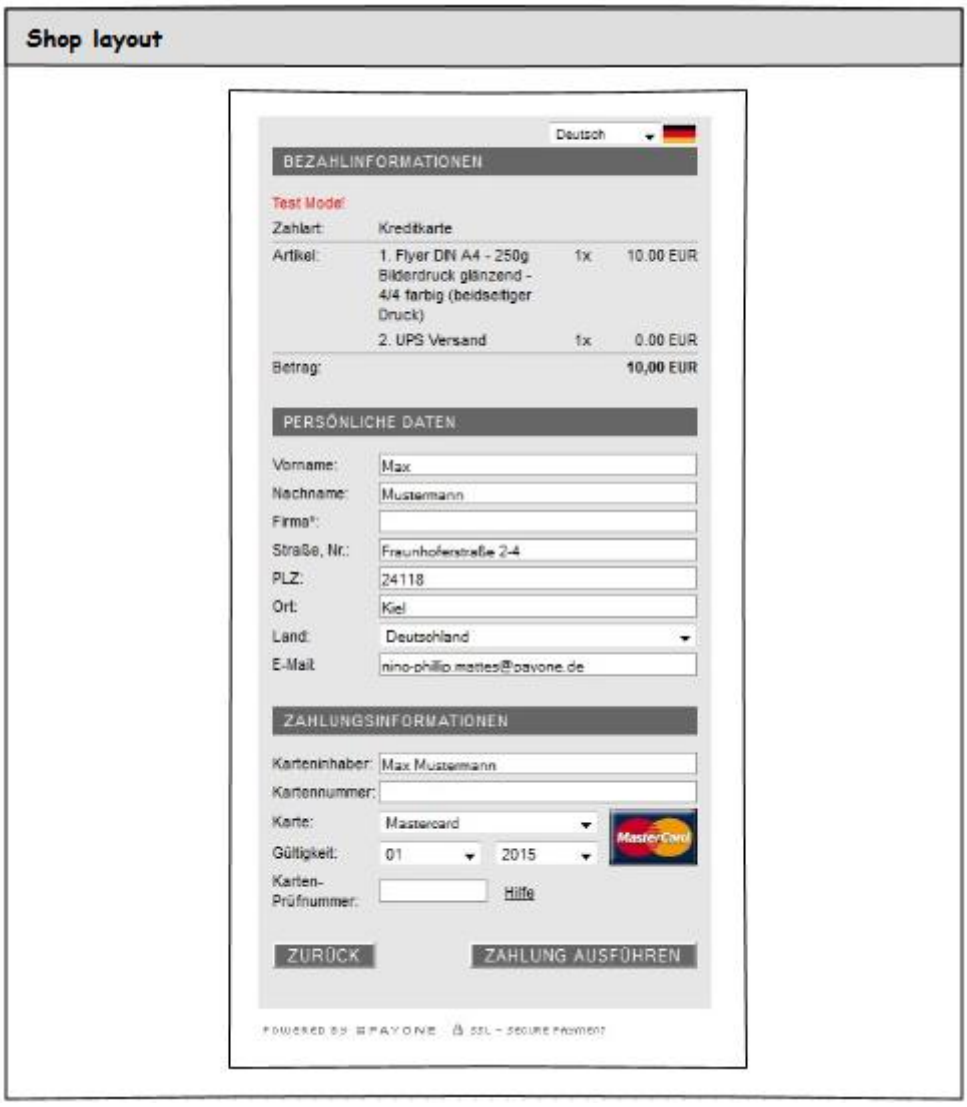
Whenever the tax and amounts including tax have to be calculated, standard methods of the shop system have to be used to ensure PAYONE's integration is conform with the shop software's standards and update-safe. However, special attention has to be paid to rounding. PAYONE's rounding precision is full cents (if transactions are performed in Euro) (cf. API descriptions, amount parameter), where usually shop systems have a higher precision.

4.2 Default checkout for redirect payments



4.3 Checkout with credit card (Channel Frontend)

Completing a checkout with transmission of credit card data through the frontend channel requires one additional step after confirming the order. The Frontent Credit Card Data Form (FCCDF) has to be embedded into the shop through and iFrame or as a redirect.



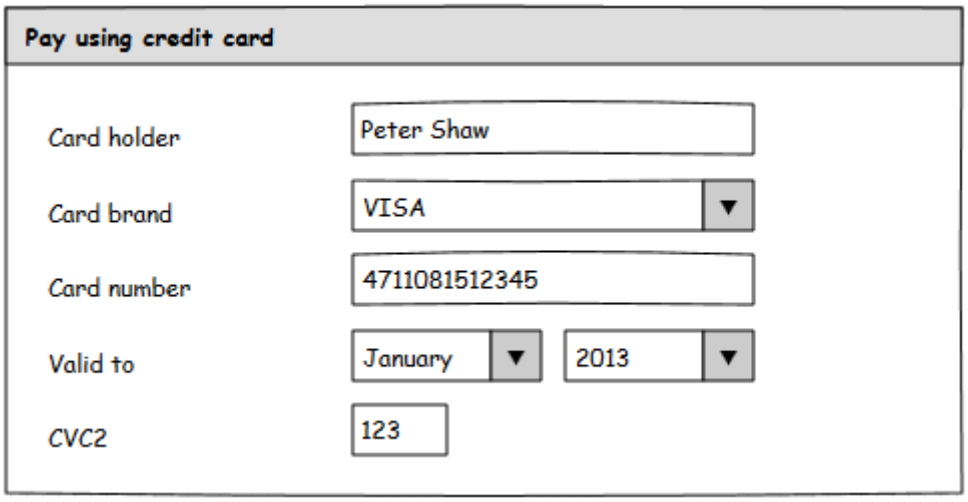
Depending on the credit card, a redirect to the customer's bank for a 3D Secure Check will follow.

The parameter `autosubmit` must not be used.

4.4 Credit Card Channel Client API

PAYONE supports two modes of credit card processing through the client API channel. Which mode might be the best to choose might depend on the merchant's requirements and on their ability to get certified according to PCI DSS. Merchants usually try to avoid certification above level SAQ A (for instance SAQ A-EP), as high certifications require e.g. penetration tests and constant monitoring of the merchant's infrastructure.

Thus, PAYONE provides the mode "hosted iFrame" for merchants aiming to get certified according to SAQ A, and the "classic" Client API mode that requires certification according to SAQ A-EP or higher. For details on PCI DSS certification and the self assessment questionaires, please refer to the PCI council's website.



Credit card payment is usually done in three steps:

1. Gather credit card data, test plausibility (classic Client API only) and submission to PAYONE either through AJAX call or hosted iFrame.
2. Preauthorize the amount on the card.
3. If the card is a 3-D Secure card, the customer is redirected to their bank's website to perform the authentication. Afterwards, they are redirected back into the shop.

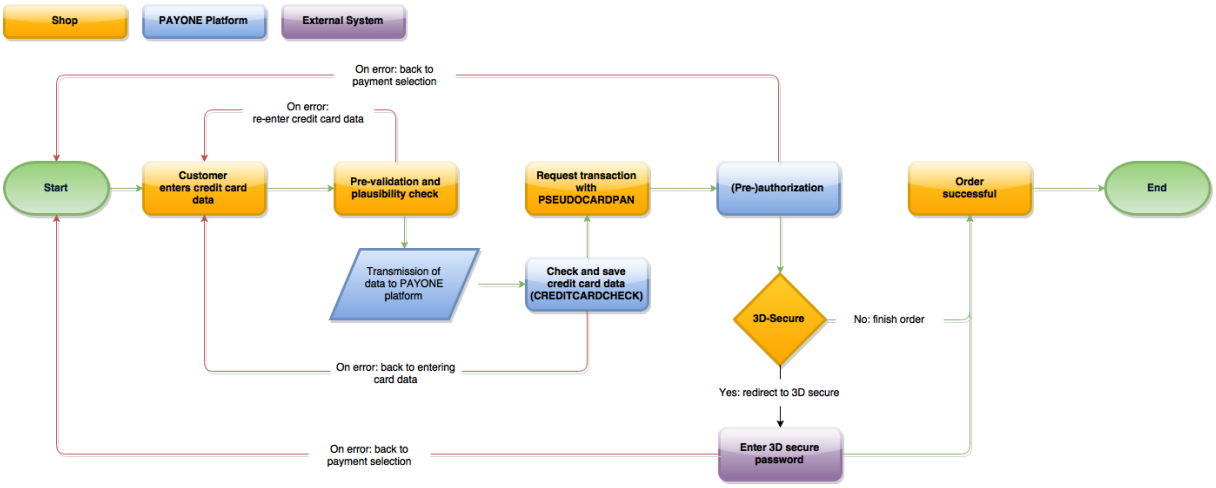
When the package is shipped, the preauthorized amount is captured using the Server API call *capture*.

To implement this process, the input fields have to be created dynamically when the customer selects to pay with their credit card. Either the hosted iFrame fields have to be created (cf. Client API documentation) or regular input fields have to be used for classic Client API mode. Depending on which mode the merchant selects, it has to be made sure that

1. all fields are filled out
2. the "valid to" date lies in the future
3. all whitespace has been removed.

After that the credit card data has to be submitted to PAYONE using either hosted iFrame or classic Client API mode with an AJAX call (request *creditcardcheck* with the parameter storecarddata=yes). A pseudo card number will be returned and has to be saved by the shop, as it is needed for the preauthorization call. The pseudo card number can be stored at the merchant's site for further reference. It is stored on PAYONE side as well.

If the customer finalizes their order, the actual transaction is started and the amount on the card is reserved. The shop transmits the order data as well as the pseudo card number to PAYONE using the Server API channel. PAYONE verifies card data and the reservable amount with the issuing bank and notifies the shop about the status of the reservation.



4.4.1 3D Secure

In case the customer's card is a 3-D Secure card, PAYONE will respond an URL instead of a confirmation. The response will always be *status=redirect* and contain a redirect URL. The customer will then be redirected to a website of the card issuing bank. Before or while the redirect takes place, the customer needs to be informed with a text similar to the following:

> Your credit card is secured with the Verified by Visa or MasterCard SecureCode procedure, that provides even more security when paying online with your credit card. In order to authenticate you will be redirected to a website of your bank. Please follow the instructions on that site and you will be redirected to the shop. After you authenticated yourself the order will be finalized.
> Or customer service will be glad to assist you, should you have any questions.

4.4.2 Redirect the customer back to the shop

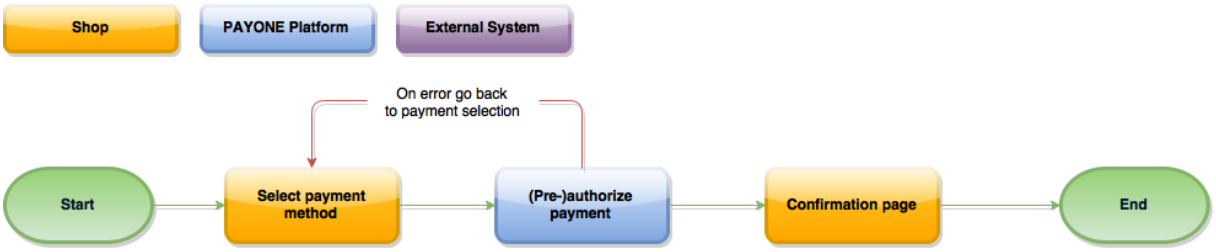
After the successful authentication the customer needs to be redirected to the success URL submitted with the (pre-)authorization request. If the authentication fails they will be redirected to the error URL. In each case a message should be displayed informing the customer about the status of their order.

4.4.3 Safety and fraud prevention

To avoid the customer just calling the success URL without entering the password it has to be made sure that either a successful response is present from the API request, or that the Transaction Status *appointed* has been transmitted by PAYONE to the shop. This HTTP callback will be sent to the shop before the customer is redirected to the shop system.

4.5 Pre-payment

A pre-payment transaction scheme looks like the following:

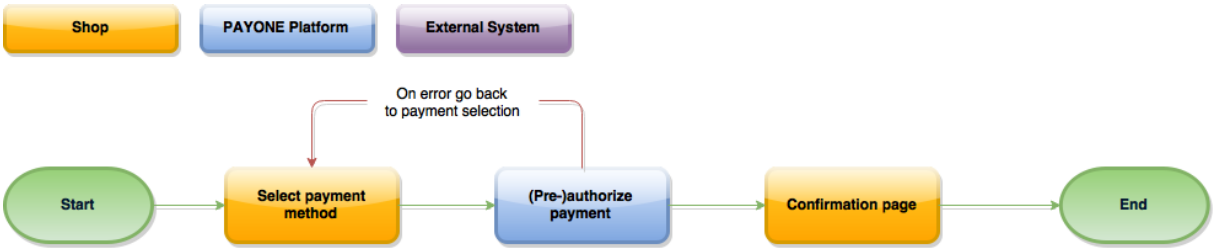


1. Open the transaction through the server API
2. Using the shop's default method, a confirmation E-Mail should be sent to the customer. This E-Mail contains the account data from the API response.
3. After the customer wired the amount to our account, PAYONE sends the Transaction Status event *paid.*
4. When shipping the order, the transaction is booked.

The first step works the same way as for credit card payment. Høwever, no card data is needed. When the transaction status *paid* arrives in the shop, it is vital to check weather the amount of the payment is greater or equal to the expected amount. If this is the case, the order can be market *paid* and further processes can be started (e.g. shipping).

4.6 Invoice and Cash on Delivery

Both payment methods have to be listed separately in the shop's frontend. However, they will be described in one section here as the process is identical for both payment methods. Cash on delivery essentially is the same as pay on invoice: the merchant sends the package and receives the money afterwards. The process on the PAYONE platform is as follows:

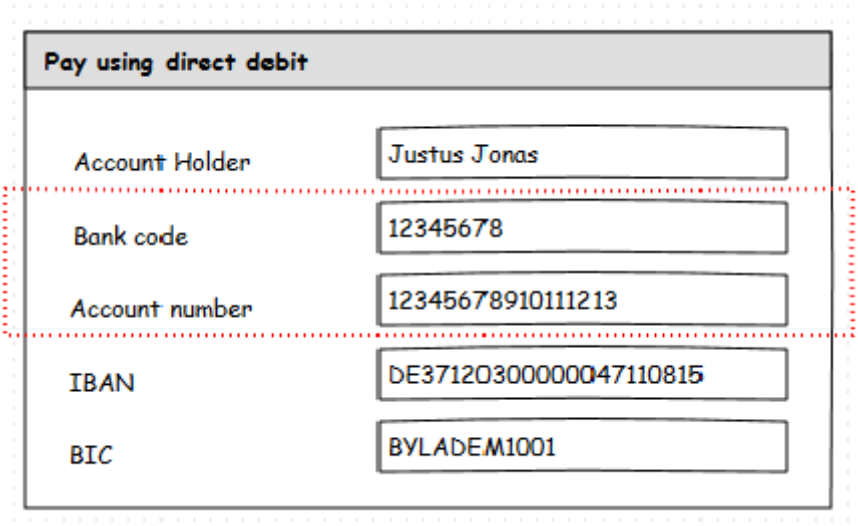


1. In checkout, the transaction is initialized useing the Server API channel
2. When shipping the items the actual booking is performed

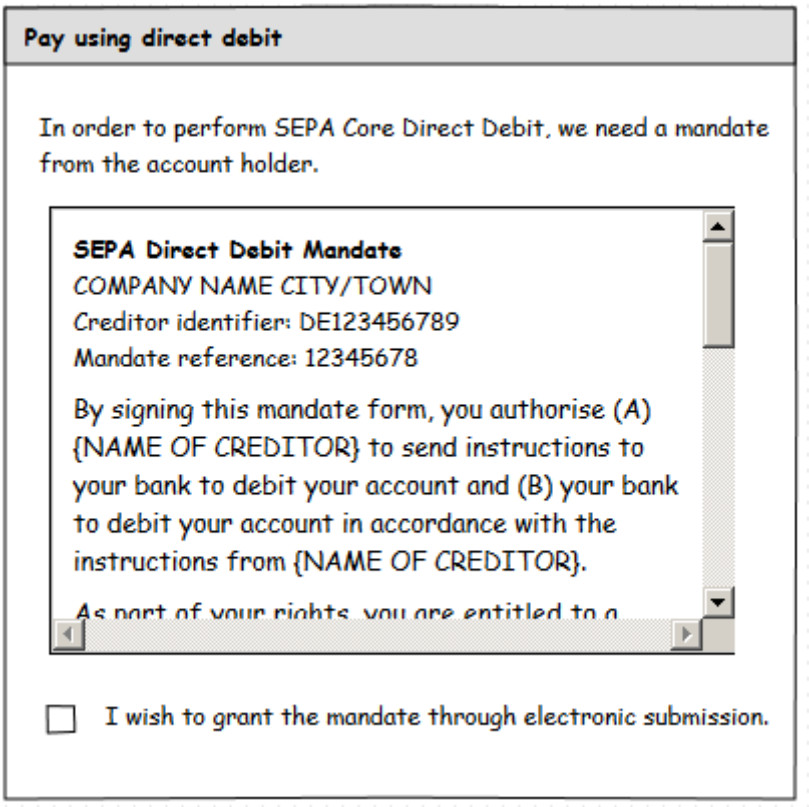
As soon as the customer or the shipping provider wires the amount, PAYONE notifies the shop using the transaction status notification *paid*.

4.7 SEPA Direct Debit

4.7.1 Step 1: Gathering the data



4.7.2 Step 2: SEPA Mandate



Please note that the current direct debit mandate text has to be used. The text shown here is just an example.

4.7.3 Step 3: Download the mandate as PDF

After the transaction is authorized the mandate will be available as a PDF download. The download link will only be displayed if mandate download is configured (see above) and the parameter mandate_identification is available.

4.7.4 Process

SEPA Core Direct Debit is processed the following way:

When the customer selects this payment method, input fields for IBAN, BIC and/or account number and bank code will be displayed according to the configuration.

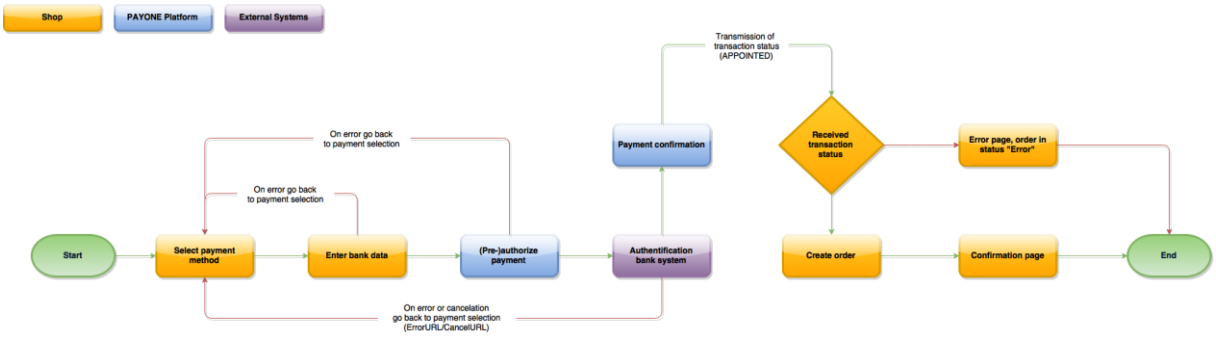
1. During checkout, the transaction will be initiated through the Server API channel
2. On shipment, the actual booking will be started.

In case of a return debit note (non-existant account, insufficient funds etc.), the PAYONE platform will send the transaction status notification *cancelation* after a couple of days.

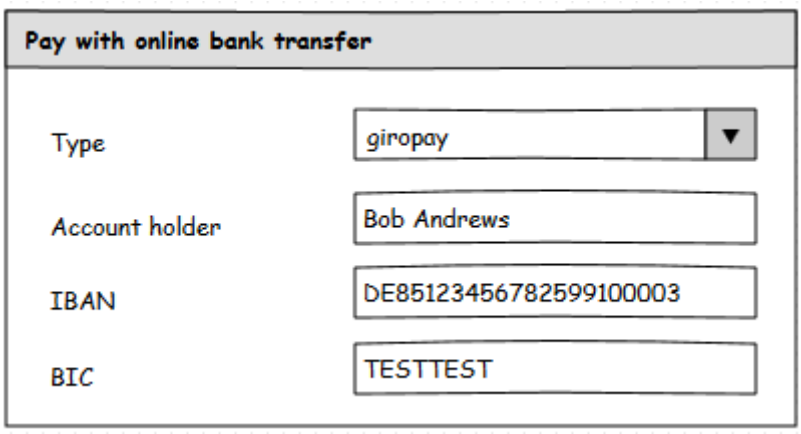
In the step after the bank data is entered, the data has to be validated. Additionally, it should be possible to perform the POS Black List check hier, which is technically the same request with one changed parameter.

Upon initiation of the transaction the bank data has to be transmitted to PAYONE. After the reservation the bank account data will be connected to the debtor record.

4.8 Online Bank Transfer

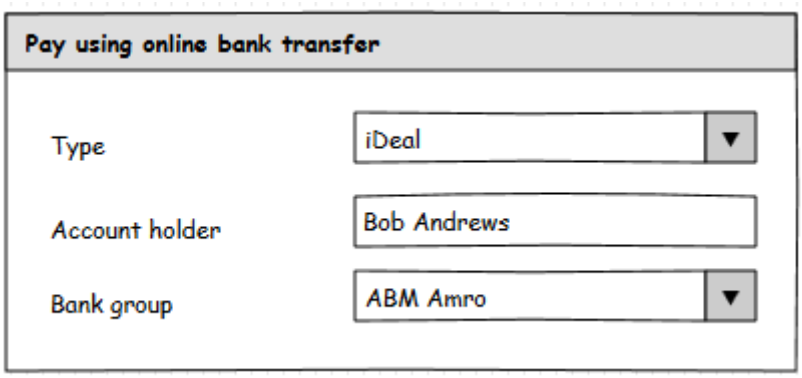


4.8.1 giropay and SOFORT.com



The online bank transfer methods *giropay* and *SOFORT.com* require a redirect to their respective pages. Since the process is identical for both providers they are described here together. However, they have to be separate payment methods in the shop's frontend. On selection of the respective payment method, input fields for IBAN and BIC have to be displayed. In the next step, a logical validation has to be performed. However, a POS blacklist check should not be performed. After checkout, the transaction will be instantiated through the Server API channel. PAYONE will return the URL to which the customer has to be redirected in order to authenticate with their bank and/or the online bank transfer provider. Upon delivery the actualy booking is triggered.

4.8.2 eps and iDeal



In contrast to online bank transfer types giropay and SOFORT.com, for eps and iDeal just to bank group type has to be selected by the customer. No IBAN or BIC is needed at the shop's checkout.

4.8.3 Przelewy24 (P24)

For the online bank transfer type Przelewy24 (P24) no additional data is needed.

4.8.4 Redirecting the buyer back in the shop (Process identical to 3D-Secure)

After successfully completing the payment process, the buyer will be redirected to the success URL supplied in the (pre-)authorization request. If the transaction was not successful, the buyer will be redirected to the error URL. When the buyer reaches the success URL an order confirmation page can be displayed.

4.8.5 Security (Process identical to 3D-Secure)

To prevent the customer manually opening the success URL without actually completing the payment process it has to be made sure that the transaction status notification with the event *appointed* was delivered to the shop. This HTTP callback will be sent as soon as the customer completes the payment process with the online bank transfer provider.

4.9 PayPal, BillSAFE, CommerzFinanz

These payment methods essentially work the same way as "normal" redirect payment methods, for instance online bank transfer. When the customer selects one of these payment methods, the shop initiates the transaction using the (pre-)authorization request. PAYONE will return a redirect URL to which the customer has to be redirected by the shop. The customer then confirms the transaction

with the third party system and will be redirected to the shop's success page if the transaction was confirmed.

4.9.1 Redirecting and Security

cf. online bank transfer.

4.9.2 PayPal ECS

PayPAL ECS (Express Checkout Shortcut) is a so-called "Predirect Payment Method". The customer is redirected to PayPal before they can enter any address data. After the customer is returned to the shop, the customers addresses will be transmitted to PAYONE by PalPal and PAYONE will send them to the shop. The shop should display an overview page stating the total cost of the order (cart value + shipping + tax) and present the customer with a final button to confirm the order. Additionally, the customer might want to select a shipping method, if more than one method is available.

The exact technical specifications are available in a separate Server API amendment.

4.10 Paydirekt

This payment method essentially work the same way as "normal" redirect payment method. When the customer selects one of these payment methods, the shop initiates the transaction using the (pre-)authorization request. PAYONE will return a redirect URL to which the customer has to be redirected by the shop. The customer then confirms the transaction with the third party system and will be redirected to the shop's success page if the transaction was confirmed.

Regarding this payment method there is a possibility to send the minimum age (`add_paydata[minimum_age]`) to the PAYONE plattform. If any minimum age is specified in the articles of the cart, the lowest minimum age had to be used. If the customer's age is below the minimum age he should be redirected to the URL specified in `add_paydata[redirect_url_after_age_verification_failure]`. If this Parameter is empty or not sent, the `errorurl`-Parameter will be used for the redirect.

The exact technical specifications are available in a separate Server API amendment.

4.10.1 Redirecting and Security

cf. online bank transfer.

4.11 Barzahlen

4.11.1 Order process

The order process is similar to an offline payment type like pre-payment or cash on delivery. The customer chooses the payment method during checkout and submits the payment on the final review page.

It is only possible to use Barzahlen with the preauthorization request. It has to be made sure that a preauthorization is sent even when authorization is set as the global default.

4.11.2 Success page

After the customer confirms the order PAYONE will return an HTML block in response to the preauthorization request. This HTML block has to be displayed on the success page of the order.

Optionally, our support can switch the HTML block for a PDF. This PDF has to be displayed on the success page. It is not possible to return both, the HTML and the PDF.

Example:

PAYONE
more than payment.

Barzahlen
Online bar bezahlen

TÜV SAARLAND Geprüftes Zahlungssystem
✓ Sicher ✓ Schnell ✓ Zuverlässig ✓ Ohne Registrierung

Der finale Schritt zu Ihrem Produkt

Gehen Sie jetzt zu einer Annahmestelle, um Ihre Bestellung schnell zu erhalten.
Drucken Sie dafür Ihren Zahlschein aus oder benutzen Sie die SMS-Variante.

(Zahlscheinkopie versendet an: max.mustermann@barzahlen.de)

Schritt 1

SMS anfordern oder **Zahlschein anzeigen**

Barzahlen mit SMS

Sie können sich den Zahlschein per SMS auf Ihr Handy schicken lassen. Klicken Sie dazu auf "SMS anfordern" und geben Sie Ihre Handynummer ein. Sie erhalten daraufhin eine SMS von Barzahlen mit Ihrem Zahlcode. In der Partnerfiliale wird der Zahlcode in die Kasse eingegeben. Sie bezahlen dann den fälligen Betrag mit Bargeld. Der Online-Shop wird **sofort** über die Bezahlung informiert und kann Ihre Bestellung weiter bearbeiten.

Barzahlen mit Zahlschein

Sie können sich den Zahlschein ausdrucken, indem Sie oben auf "Zahlschein anzeigen" klicken. Gehen Sie danach mit dem ausgedruckten Zahlschein in die nächste Partnerfiliale und lassen Sie den Zahlschein an der Kasse einscannen. Sie bezahlen dann den fälligen Beitrag mit Bargeld. Der Online-Shop wird **sofort** über die Bezahlung informiert und kann Ihre Bestellung weiter bearbeiten.

Schritt 2

Ihre nächsten Barzahlen-Partnerfilialen:

dm 1,2km: **dm-drogerie markt, Massower Straße 5, 10315 Berlin**
Geöffnet Mo-Sa 08:00-20:00 Uhr

real,- 1,4km: **real,- Markt, Frankfurter Allee 113-117, 10365 Berlin**
Geöffnet Mo-Sa 8:00-22:00 Uhr

dm 1,6km: **dm-drogerie markt, Frankfurter Allee 111, 10247 Berlin**
Geöffnet Mo-Sa 09:30-21:00 Uhr

Sie suchen eine Filiale? - Ihre Lösung ist unser Barzahlen-Filialfinder

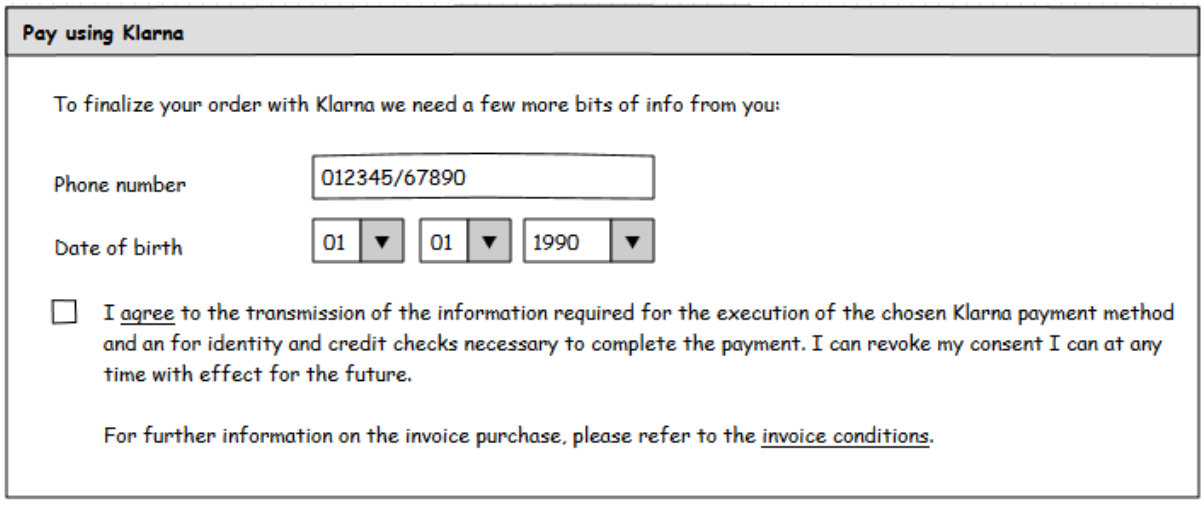
Die nächste Barzahlen-Partnerfiliale in Ihrer Umgebung finden Sie ganz einfach mit unserem Filialfinder unter www.barzahlen.de/filialfinder oder mit unserer Barzahlen-App.

4.11.3 Dokumentation

Die Beschreibung der Schnittstelle ist nicht Bestandteil dieses Dokuments und liegt als gesondertes Dokument vor.

4.12 klarna

4.12.1 Klarna Invoice



In order to use Klarna as a payment method, some additional pieces of information have to be requested during checkout, depending on the buyer's country. In Germany, for instance, the buyer must agree to the transmission of their data to a third party, in this case Klarna. Klarna's specifications for integrating the terms and conditions can be found here, for instance:

https://developers.klarna.com/en/de+php/kpm/consumer-terms-and-conditions

Some of the data required by Klarna might be available from earlier steps of the checkout. These fields should not be displayed, but should still be transmitted to PAYONE. The rest of the process is identical to "classic" invoice or cash on deliver. However, only preauthorization is possible as of now.

Please note that the date of birth always has to be transmitted.

4.12.2 Part payment

Integrating part payment is very similar to integrating Klarna invoice. However, every campaign has to be represented as a single payment method in checkout. The merchant thus has to duplicate Klarna Part Payment payment methods and assign them names and campaing codes.

4.13 Payolution

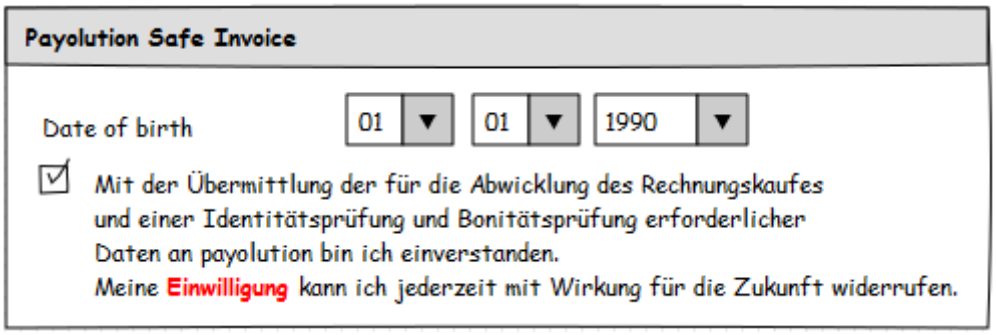
Caution

As of now, all Payolution payment methods must be preauthorization transactions!

Info
Payolution offers B2B transactions which require additional parameters, see list in Server-API Documentation. B2B Transactions should be configurable in the shop's backend. If B2B is enabled, the customer must be given the opportunity to enter the companies VAT ID and Trade Registry Number during checkout, if the transaction is deemed to be a B2B transaction, e.g. when a company has been entered as billing address. For a B2B transaction the date of birth should not be queried from the user. These parameters have to be parsed into the corresponding add_paydata parameters, and add_paydata[b2b] must be set to "yes".

Info
While the `add_paydata[action]=pre_check` request is mandatory and should be submitted once the customer chose a Payolution payment method, the `add_paydata[action]=refund_announcement` and `add_paydata[action]=fulfillment_delay` requests are optionally to be implemented in standard shop integrations.

4.13.1 Safe Invoice

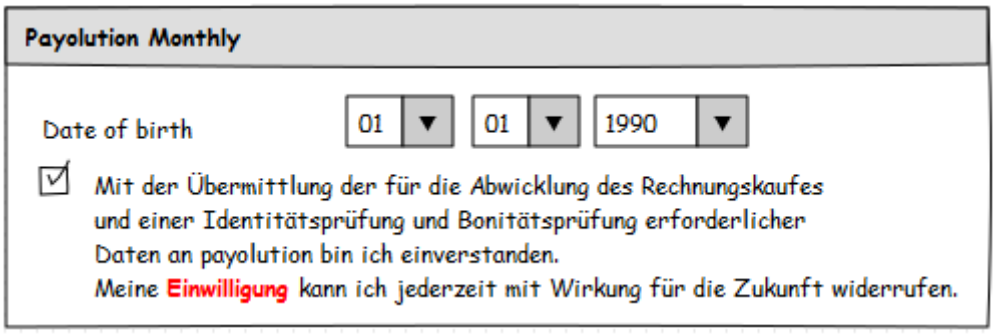


If it hasn't already been entered, when Payolution Safe Invoice is selected the date of birth has to be queried from the customer (cf. Klarna in this document). The word "Einwilligung" marked red in the figure above has to be a link. This link has to open a modal window/overlay where the privacy agreement provided by Payolution is displayed. The source of the privacy agreement is https://payment.payolution.com/payolution-payment/infoport/dataprivacydeclaration?mId=%s where %s is the base64 encoded company name as configured in the backend. The customer must not know at any time that this document is delivered by a Payolution server.

Info

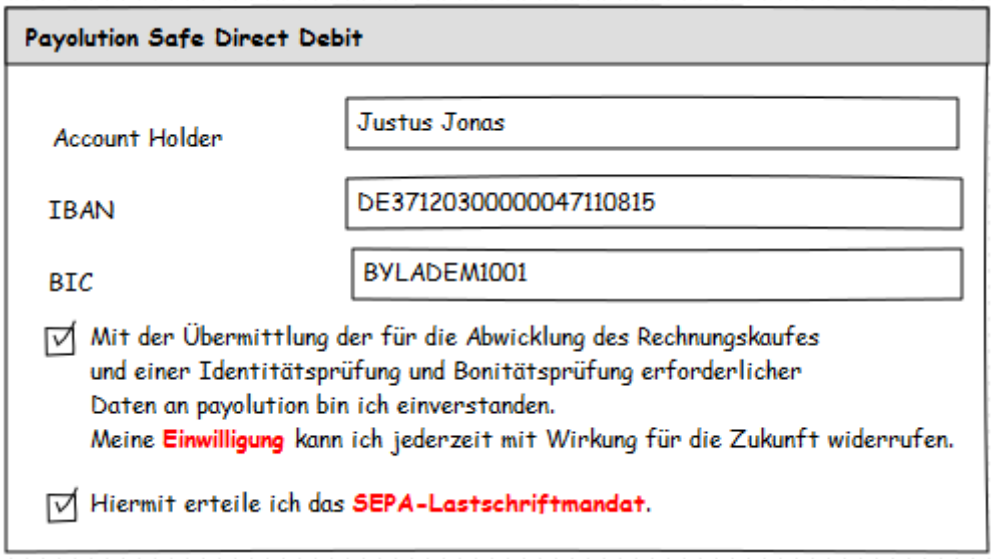
Payolution will return a reference which has to be printed as the reason for payment on the final invoice for the customer. It will be returned upon any successful authorization or preauthorization request and is stored in the parameter `add_paydata[clearing_reference]`. It has to be made available in the Shop's backend and database as well, for instance in an order detail page, as it might be needed for Customer Care or third party systems.

4.13.2 Monthly



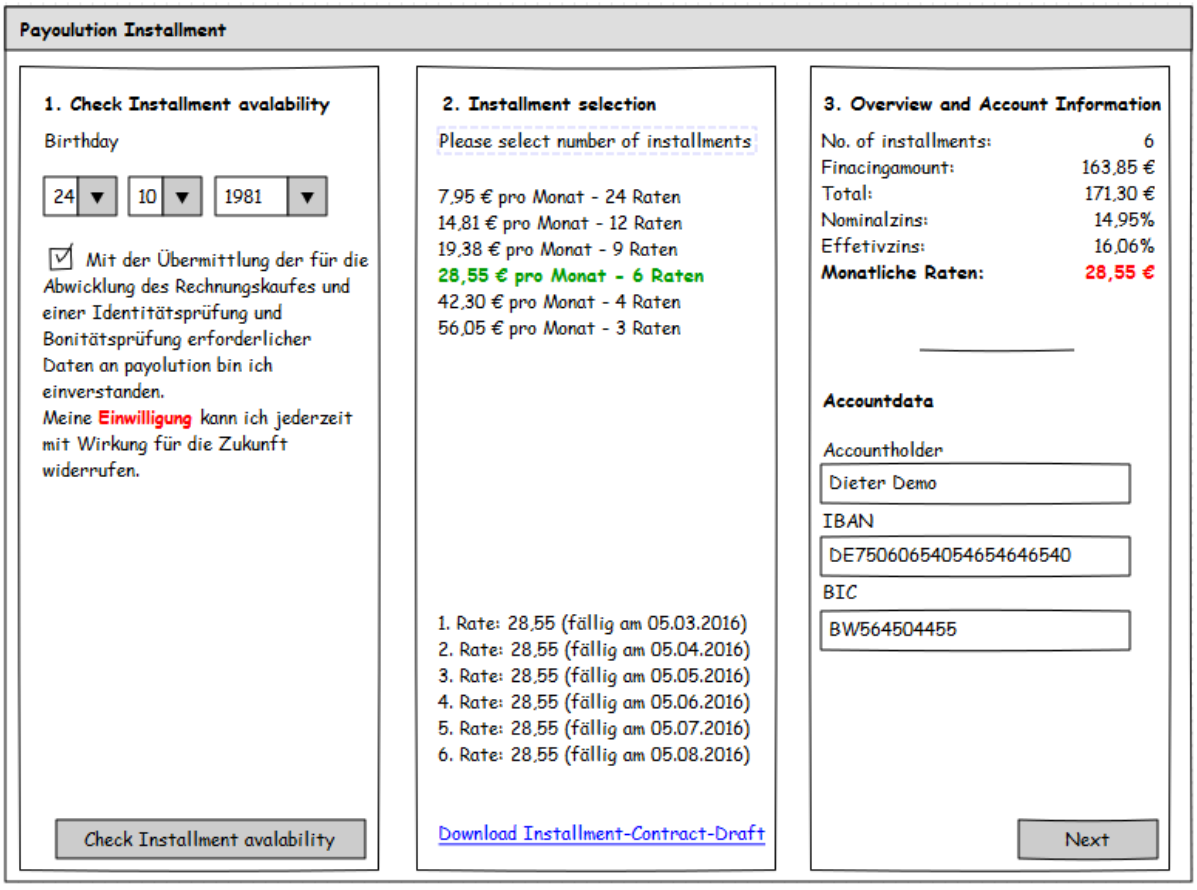
The same principles of Payolution Safe Invoice apply. The date of birth has to be queried from the user if not available from previous checkout steps.

4.13.3 Safe Debit



The same principles of Payolution Safe Invoice apply. Additionally, the bank account information has to be queried from the customer. All fields, name of the account holder, IBAN, and BIC are necessary. The customer is required to acknowledge the privacy statement and the SEPA mandate, which is available through the URL https://payment.payolution.com/payolution-payment/infoport/sepa/mandate.pdf and can be displayed either in a modal window, overlay, or similar; or in a separate browser window. The date of birth has to be queried from the user if not available from previous checkout steps.

4.13.4 Installment



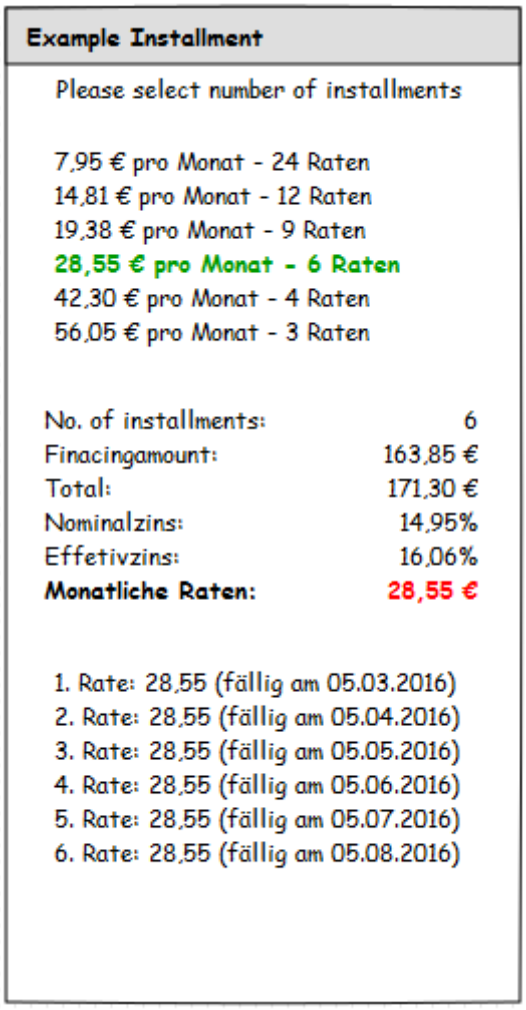
Guiding the customer through the Payolution financing payment method is a three step sub-process that should entirely be conducted using AJAX calls (or similar) to avoid refreshing the entire payment page. Which a click on "Check Installment availability" a call should be made to the shop that issues a `genericPayment` request with the `add_paydata[action]=pre_check` parameter which will return whether installment is available for this customer or not and will return a reference `workorderid`, cf. Server API description for details.

In the second step, a `genericPayment` request with the parameter `add_paydata[action]=calculation` has to be submitted to obtain the different payment plan possibilities. The different plans will be returned in `add_paydata[PaymentDetails_n_*]` parameters, where `n` is a numeric identifier of the payment plan. Ideally, the duration and the monthly payment are displayed to the customer for them to select. Also, a link to the Installment Contract Draft has to be provided which has to be displayed in a modal window or overlay. It has to be obtained from the `add_paydata[PaymentDetails_n_StandardCreditInformationUrl]` parameter by using the appropriate credentials provided in the configuration interface for HTTP authentication, i.e. `merchantname-installment` with the corresponding shared secret. The credentials must not be visible to the customer in any way. In the third column, all details about the payment plan have to be displayed. In the following authorization/preauthorization, the `workorderid` as well as the duration of the customer selected payment plan have to be submitted, cf. Server API description for details.

All other processes work similar to Payolution Invoice.

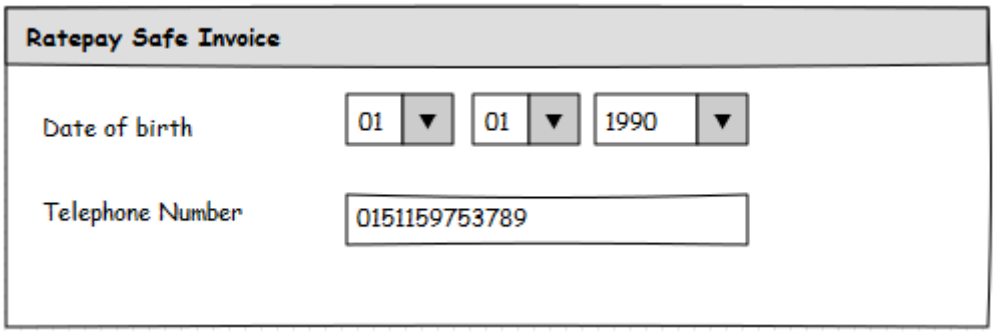
4.13.4.1 Switzerland

4.13.4.2 Article



4.14 RatePay

4.14.1 Safe Invoice



If it hasn't already been entered, when RatePay Safe Invoice is selected the telephone number and the date of birth has to be queried from the customer (cf. Klarna in this document). The minimum age for orders with RatePay is 18 years. This has to be actively checked and enforced. Additionally, the telephone number is mandatory and has to be queried from the user if it's not available from a previous checkout step.

RatePay requires a device fingerprint (DFP) for every (pre-)authorization. This fingerprint has to be unique per customer + checkout + device. It should be generated as early as possible, but only once per session, i.e. on the basket page, when entering the checkout, or on the payment selection page the latest. One session ends as soon as the checkout is completed and a new `deviceIdentToken` and fingerprint have to be created. If the customer can repeat the purchase, a new token has to be generated as well. Token and TXID are in a 1:1 relationship.

In order to create the device fingerprint, two tokens have to be obtained:

1. The `deviceIdentSId` which is defined by RatePay and available through the Profile request, which will be issued once in the backend of the shop system.

2. A unique `deviceIdentToken` which can for instance consist of a rather static and a (semi-)unique component, for instance a customer ID and a `uniqid()` or `microtime()` hashed as SHA384 or MD5.

Then, some static JavaScript has to be integrated into the DOM of the checkout page:

```
<script language="JavaScript">
var di =
{t:'___$deviceIdentToken___',v:'___$deviceIdentSId___',l:'Checkout'}
;
</script>
<script type="text/javascript"
src="//d.ratepay.com/___$deviceIdentSId___/di.js"></script>
<noscript><link rel="stylesheet" type="text/css"
href="//d.ratepay.com/di.css?t=___$deviceIdentToken___&v=___$deviceI
dentSId___&l=Checkout"></noscript>
<object type="application/x-shockwave-flash"
data="//d.ratepay.com/___$deviceIdentSId___/c.swf" width="0"
height="0">
<param name="movie"
value="//d.ratepay.com/___$deviceIdentSId___/c.swf" />
<param name="flashvars"
value="t=___$deviceIdentToken___&v=___$deviceIdentSId___"/><param
name="AllowScriptAccess" value="always"/>
</object>
```

Where `___$deviceIdentSId___` and `___$deviceIdentToken___` have to be replaced by the respective values by appropriate methods.

Transaction reference
RatePay will return a reference which has to be printed as the reason for payment on the final invoice for the customer. This reference is returned after `authorization`, `preauthorization`, and `capture` requests in the parameter `add_paydata[clearing_reference]`.

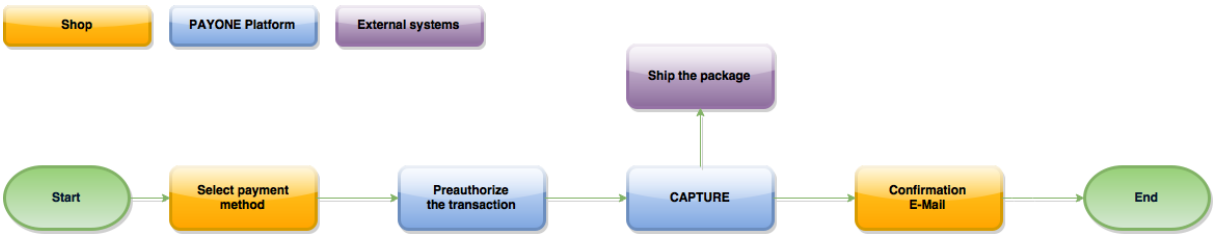
5 Functions in the backend

5.1 Capturing a preauthorized amount (capture)

After successfully initiating a transaction with a preauthorization request, the amount needs to be booked to fulfill the order. Either this can be performed by an external ERP system, or through the shop software. To enable a merchant without ERP software to fulfill transactions, there has to be the possibility to capture transactions from within the shop system. Here, PAYONE supports two cases:

Case 1: Capture the entire amount

The entire amount should be captured. For this, a *capture* request is sent to PAYONE through the server API, containing the transaction ID from the preauthorization, the amount and the article list.

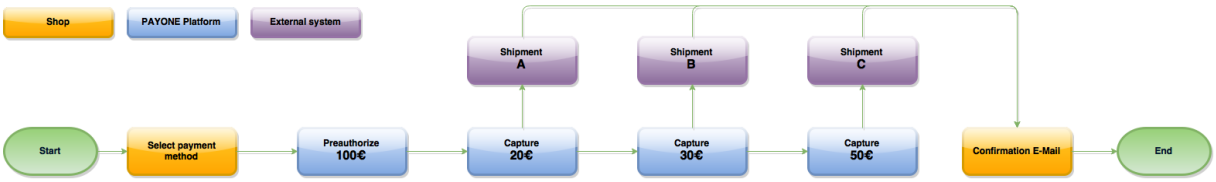


Case 2: Partial delivery (Multiple captures)

This case only applies to the payment methods credit card, direct debit, invoice, pre-payment, and cash on delivery. All other payment methods have to be captured in one request, since multiple capture is not supported.

When delivering an order in multiple packets there can be multiple captures. An order of 100 EUR has to be performed in the following way:

1. preautorization, PAYONE returns txid 123
2. Shipment A (value 20 EUR): capture 20 EUR on txid 123
3. Shipment B (value 30 EUR): capture 30 EUR on txid 123
4. Shipment C/final shipment (value 50 EUR): capture 50 EUR on txid 123



5.2 Debit on a transaction

The PAYONE platform can handle debit notes automatically, without storing credit card or account data in the shop. This can be done the same way as a capture, for instance in an ERP system. However, also merchants only using the shop system must be able to refund transactions.

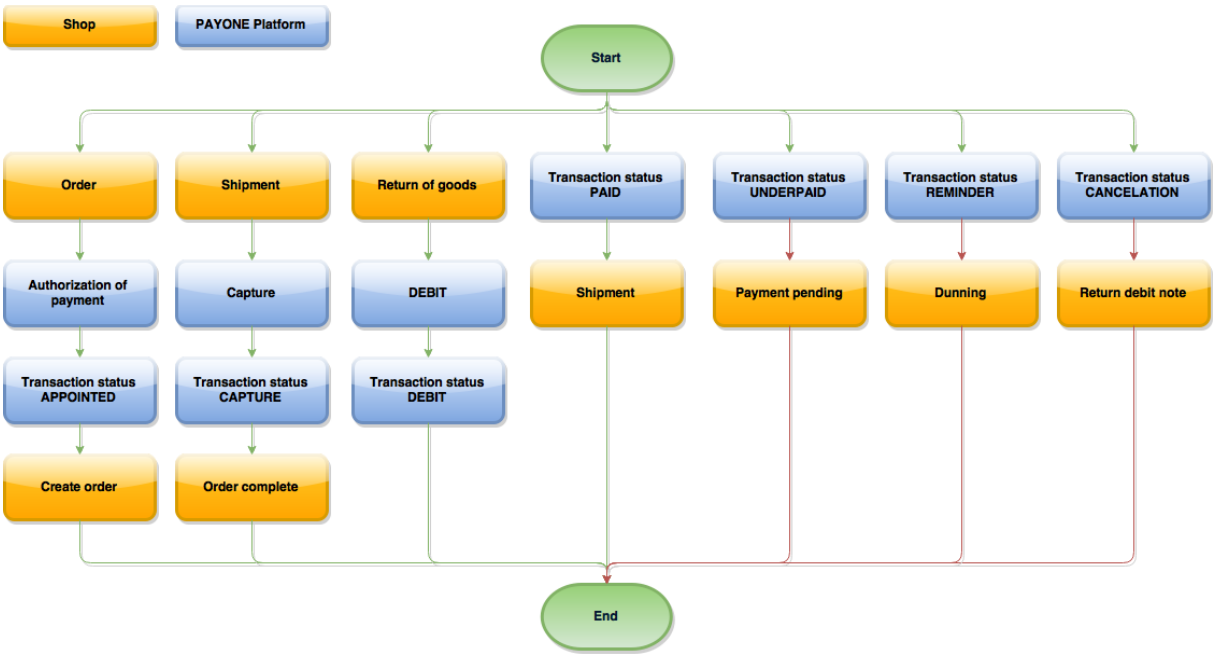
Similar to the capture functionality there must be a possibility to refund transactions. Full refunds must be possible as well as refunding selected positions of an order. Additionally, the merchant must be able to refund arbitrary amounts, however they must not exceed the total amount of the order.

5.3 Processing asynchronous notifications: "transaction status"

The PAYONE platform sends for each transaction event a asynchronous status message, the "transaction status". This is a HTTP callback sent to a specific shop URL as a POST request submitting key-value-pairs. The transaction status notifications covers notifications about initialized transactions, incoming payments, dunning, and much more.

Furthermore, the transaction status notifications have to be used as a security measure for redirect payment methods (for instance online bank transfer or credit card with 3D secure). Also, the transaction status notifications must be used to note and process incoming payments for invoice and pre-payment transactions and for cancelations.

All events have to be logged following the logging guidelines specified in this document.



5.4 Information

The info page contains an PAYONE info page hosted by PAYONE and made available through an iFrame.

5.5 Logs

The logs are designed to be a help and control resource for merchants, and they help our Technical Support as well as the merchant to trace back errors.

Logging has to be disabled per default. The merchant explicitly has to turn on logging.

5.5.1 Security

If the server gets compromised the logged data should be unusable. For this, all personal and payment data have to be masked before they are written to the log.

Example:

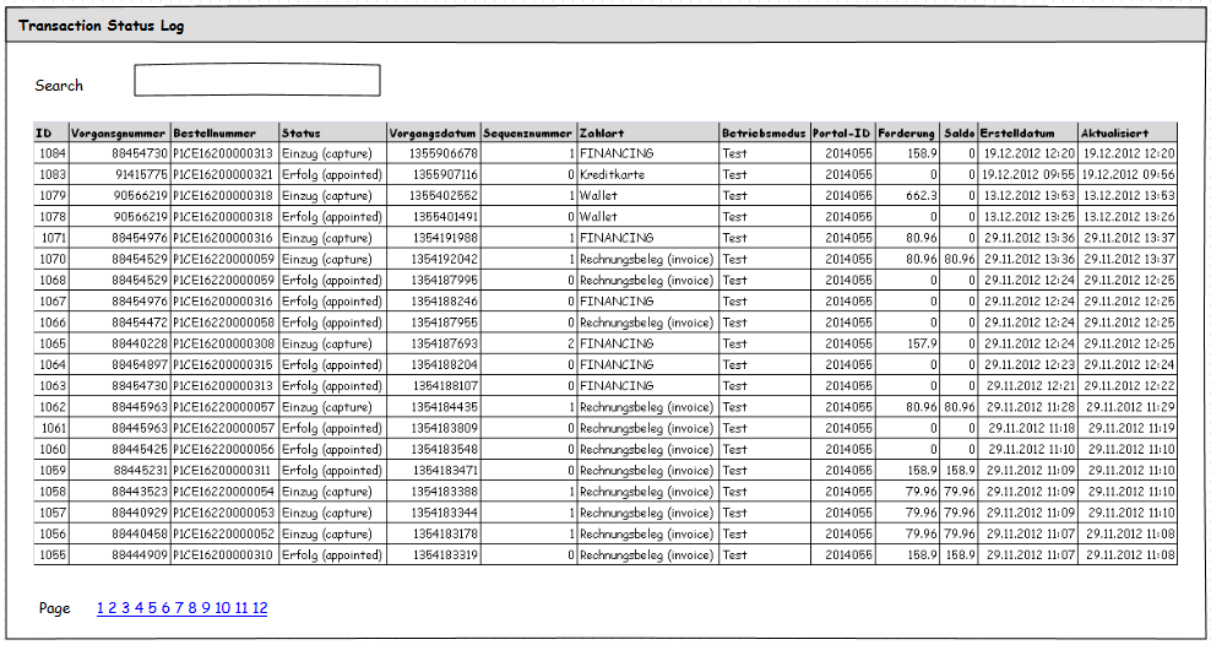
Original	Masked
cardpan=4111111111111111	cardpan=4111xxxxx1111
iban=DE85123456782599100003	iban=DE85xxxxxxxxxxxxxx003
street=Haupstrasse 1	street=Hxxxxxxxxxxx1

Readable logs must never be written in publically readable directories!

5.5.2 Transaction status log

In this log, all transaction status notification should be stored. Additionally, a search and filter function is needed.

5.5.2.1 Overview



Transaction Status Log

Search

ID	Vorgangsnummer	Bestellnummer	Status	Vorgangsdatum	Sequenznummer	Zahlart	Betriebsmodus	Portal-ID	Forderung	Saldo	Erstelldatum	Aktualisiert
1084	88454730	PICE16200000313	Einzug (capture)	1355906678	1	FINANCING	Test	2014055	158.9	0	19.12.2012 12:20	19.12.2012 12:20
1083	91415775	PICE16200000321	Erfolg (appointed)	1355907116	0	Kreditkarte	Test	2014055	0	0	19.12.2012 09:55	19.12.2012 09:56
1079	90566219	PICE16200000318	Einzug (capture)	1355402552	1	Wallet	Test	2014055	662.3	0	13.12.2012 13:53	13.12.2012 13:53
1078	90566219	PICE16200000318	Erfolg (appointed)	1355401491	0	Wallet	Test	2014055	0	0	13.12.2012 13:25	13.12.2012 13:26
1071	88454976	PICE16200000316	Einzug (capture)	1354191988	1	FINANCING	Test	2014055	80.96	0	29.11.2012 13:36	29.11.2012 13:37
1070	88454529	PICE16220000059	Einzug (capture)	1354192042	1	Rechnungsbeleg (invoice)	Test	2014055	80.96	80.96	29.11.2012 13:36	29.11.2012 13:37
1068	88454529	PICE16220000059	Erfolg (appointed)	1354187995	0	Rechnungsbeleg (invoice)	Test	2014055	0	0	29.11.2012 12:24	29.11.2012 12:25
1067	88454976	PICE16200000316	Erfolg (appointed)	1354188246	0	FINANCING	Test	2014055	0	0	29.11.2012 12:24	29.11.2012 12:25
1066	88454472	PICE16220000058	Erfolg (appointed)	1354187955	0	Rechnungsbeleg (invoice)	Test	2014055	0	0	29.11.2012 12:24	29.11.2012 12:25
1065	88440228	PICE16200000308	Einzug (capture)	1354187693	2	FINANCING	Test	2014055	157.9	0	29.11.2012 12:24	29.11.2012 12:25
1064	88454897	PICE16200000315	Erfolg (appointed)	1354188204	0	FINANCING	Test	2014055	0	0	29.11.2012 12:23	29.11.2012 12:24
1063	88454730	PICE16200000313	Erfolg (appointed)	1354188107	0	FINANCING	Test	2014055	0	0	29.11.2012 12:21	29.11.2012 12:22
1062	88445963	PICE16220000057	Einzug (capture)	1354184435	1	Rechnungsbeleg (invoice)	Test	2014055	80.96	80.96	29.11.2012 11:28	29.11.2012 11:29
1061	88445963	PICE16220000057	Erfolg (appointed)	1354183809	0	Rechnungsbeleg (invoice)	Test	2014055	0	0	29.11.2012 11:18	29.11.2012 11:19
1060	88445426	PICE16220000056	Erfolg (appointed)	1354183548	0	Rechnungsbeleg (invoice)	Test	2014055	0	0	29.11.2012 11:10	29.11.2012 11:10
1059	88445231	PICE16200000311	Erfolg (appointed)	1354183471	0	Rechnungsbeleg (invoice)	Test	2014055	158.9	158.9	29.11.2012 11:09	29.11.2012 11:10
1058	88443523	PICE16220000054	Einzug (capture)	1354183388	1	Rechnungsbeleg (invoice)	Test	2014055	79.96	79.96	29.11.2012 11:09	29.11.2012 11:10
1057	88440929	PICE16220000053	Einzug (capture)	1354183344	1	Rechnungsbeleg (invoice)	Test	2014055	79.96	79.96	29.11.2012 11:09	29.11.2012 11:10
1056	88440458	PICE16220000052	Einzug (capture)	1354183178	1	Rechnungsbeleg (invoice)	Test	2014055	79.96	79.96	29.11.2012 11:07	29.11.2012 11:08
1055	88444909	PICE16200000310	Erfolg (appointed)	1354183319	0	Rechnungsbeleg (invoice)	Test	2014055	158.9	158.9	29.11.2012 11:07	29.11.2012 11:08

Page 1 2 3 4 5 6 7 8 9 10 11 12

5.5.2.2 Detail

Transaction Status Log entry details

aid	19551
balance	0
clearing_bankcode	0
clearingtype	rec
currency	EUR
customerid	3
key	b1c690c1a20492a1aca7cc3f166372a4
mode	test
portalid	2014055
receivable	0
reference	P1CE16220000058
sequencenumber	0
txaction	appointed
txid	88454472
txtime	1354187955
userid	28365215

5.5.3 API log

In this log, all requests to and responses from the PAYONE platform should be logged. A filter and search function should be available as well.

5.5.3.1 Overview

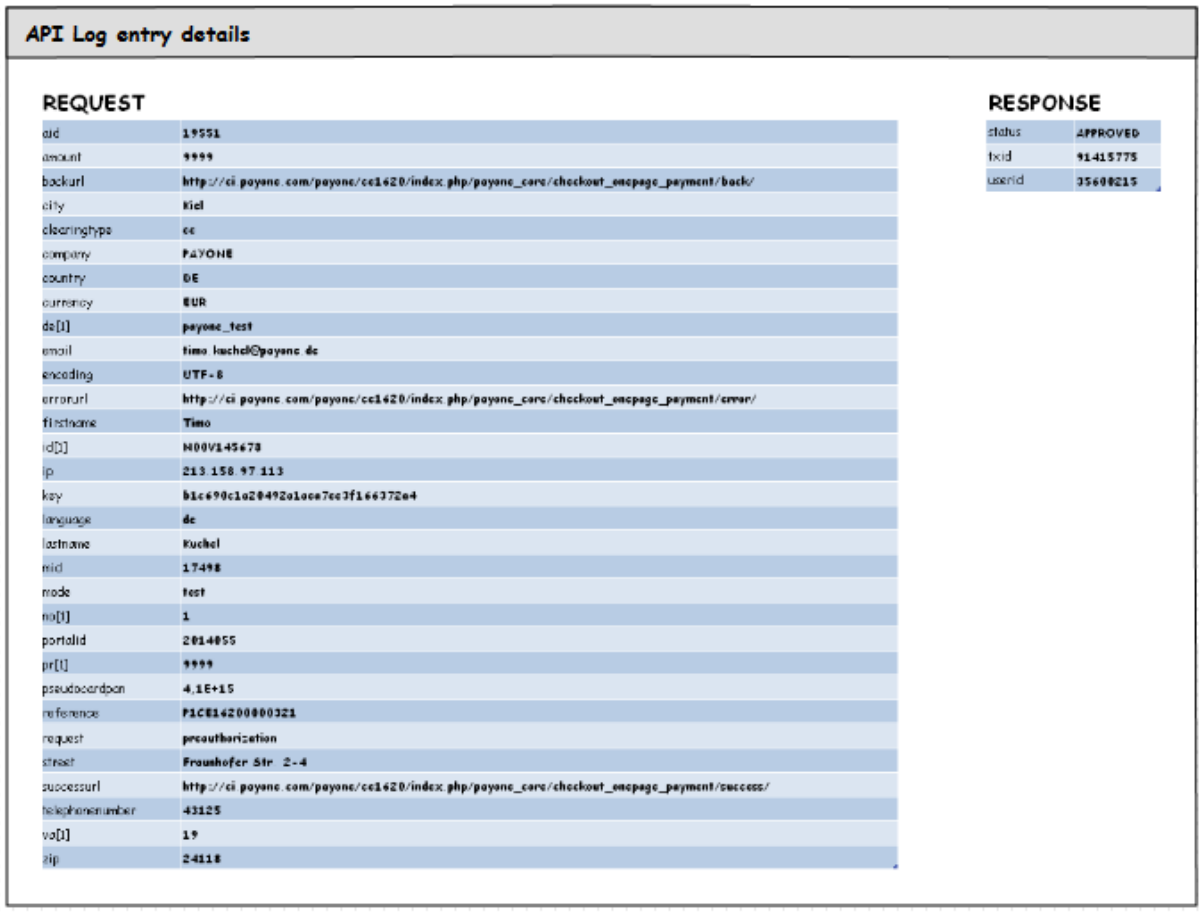
API Log

Search: Seach for...

ID	Request	Response	Modus	Merchant - ID	Portal ID	Erstelldatum
1520	Vorautorisierung (preauthorization)	Bestätigt (approved)	Test	17498	2014055	19.12.2012 09:51
1519	Adressprüfung (addresscheck)	Zulässig (valid)	Test	17498	2014055	19.12.2012 09:50
1518	Einzug (capture)	Bestätigt (approved)	Test	17498	2014055	19.12.2012 09:44
1514	Vorautorisierung (preauthorization)	Weiterleitung (redirect)	Test	17498	2014055	13.12.2012 14:30
1513	Adressprüfung (addresscheck)	Zulässig (valid)	Test	17498	2014055	13.12.2012 14:30
1512	Vorautorisierung (preauthorization)	Weiterleitung (redirect)	Test	17498	2014055	13.12.2012 14:11
1511	Adressprüfung (addresscheck)	Zulässig (valid)	Test	17498	2014055	13.12.2012 14:11
1510	Einzug (capture)	Bestätigt (approved)	Test	17498	2014055	13.12.2012 13:42
1509	Vorautorisierung (preauthorization)	Weiterleitung (redirect)	Test	17498	2014055	13.12.2012 13:24
1508	Adressprüfung (addresscheck)	Zulässig (valid)	Test	17498	2014055	13.12.2012 13:24
1507	Adressprüfung (addresscheck)	Zulässig (valid)	Test	17498	2014055	13.12.2012 13:18
1506	Autorisierung (authorization)	Bestätigt (approved)	Test	17498	2013866	07.12.2012 09:24
1505	Adressprüfung (addresscheck)	Zulässig (valid)	Test	17498	2014055	07.12.2012 09:24
1498	Rechnungsdownload (getinvoice)	Fehler (error)	Test	17498	2014055	29.11.2012 14:18
1497	Einzug (capture)	Bestätigt (approved)	Test	17498	2014055	29.11.2012 13:27
1496	Einzug (capture)	Bestätigt (approved)	Test	17498	2014055	29.11.2012 13:26
1494	Vorautorisierung (preauthorization)	Weiterleitung (redirect)	Test	17498	2014055	29.11.2012 12:24
1493	Adressprüfung (addresscheck)	Zulässig (valid)	Test	17498	2014055	29.11.2012 12:23

Page 1 2 3 4 5 6 7 8 9 10 11 12

5.5.3.2 Detail



API Log entry details

REQUEST

aid	19551
amount	9999
backurl	http://ci.payone.com/payone/cc1620/index.php/payone_core/checkout_onepage_payment/back/
city	Kiel
clearingtype	cc
company	PAYONE
country	DE
currency	EUR
de[1]	payone_test
email	timo.kuchel@payone.de
encoding	UTF-8
errorurl	http://ci.payone.com/payone/cc1620/index.php/payone_core/checkout_onepage_payment/error/
firstname	Timo
id[1]	M00V145678
ip	213.158.97.113
key	b1c690c1a20492a1ace7ec3f166372e4
language	de
lastname	Kuchel
mid	17498
mode	test
no[1]	1
portalid	2014055
pr[1]	9999
pseudocardpan	4.1E+15
reference	P1CE14200000321
request	preauthorization
street	Fraunhofer Str. 2-4
successurl	http://ci.payone.com/payone/cc1620/index.php/payone_core/checkout_onepage_payment/success/
telephonenumber	43125
va[1]	19
zip	24118

RESPONSE

status	APPROVED
txid	91415775
userid	35600215

6 Integration parameters

For each request, the following parameters have to be transmitted to PAYONE

Parametername	Beschreibung	Beispiel
solution_name	name of the software	oxid
solution_version	version of the software	4.5.6
integrator_name	name of the integrator	fatchip
integrator_version	version of the integration (Modul/Extension/App/Cartridge/etc.)	1.1.8