

# Special remarks - Klarna Payments

## General Notes

Payment Type	Countries	Currency
Invoice ("Pay later")	<ul style="list-style-type: none"> <li>Austria</li> <li>Denmark</li> <li>Finland</li> <li>Germany</li> <li>Netherlands</li> <li>Norway</li> <li>Sweden</li> <li>Switzerland</li> </ul>	<ul style="list-style-type: none"> <li>Euro</li> <li>Danish crowns</li> <li>Norwegian crowns</li> <li>Swedish crowns</li> <li>Swiss Francs</li> </ul>
Installments ("Slice it")		
Direct Debit ("Pay now")		

B2B transactions are currently only supported in Finland, Germany, Norway and Sweden.

## API-Requests - Special Notes / Deviations

### Overview of Deviations

- Klarna requires more specific customer data - depending on country of customer
- Status "PENDING" has to be processed
  - for API-response "PENDING" implementation of API-version 3.11 or higher is required
  - for TransactionStatus "pending" implementation of notify-version 7.6 or higher is required
- Request "capture" has to use parameter "capturemode"
- Request "debit" has to use parameter "settleaccount"
- add\_paydata and workorderid have to be used for update of shopping cart

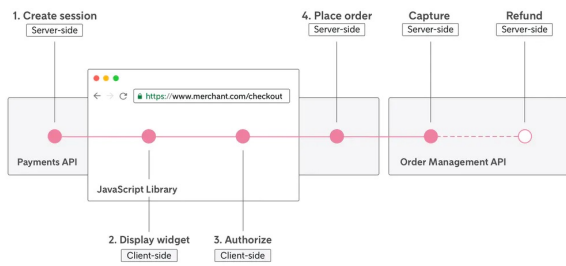
### Clearingtype / Clearingsubtype

clearingtype	financingtype	
fnc	KIS	Klarna "Slice It" (Installments)
fnc	KIV	Klarna "Pay Later" (Invoice)
fnc	KDD	Klarna "Pay Now" (Direct Debit)
fnc	KBT	Klarna "Pay Now" (Direct Bank Transfer)

### Summary

Enabling your customers to pay via Klarna requires you to integrate the Klarna payment method into your checkout process. In order to get an overview of the relationship and interaction on your checkout please see the following diagram. (This diagram is adopted from the official Klarna documentation: [Klarna Docs - Offering Klarna payment methods to your customers](#))

- 1 General Notes
- 2 API-Requests - Special Notes / Deviations
  - 2.1 Overview of Deviations
  - 2.2 Clearingtype / Clearingsubtype
  - 2.3 Summary
    - 2.3.1.1 Create a New Session
      - 2.3.1.1.1 Optional - Update the Session
    - 2.3.1.2 Handle Token
    - 2.3.1.3 Finalize the Checkout ((Pre-) Authorization)
  - 2.3.2 Klarna Checkout Processing Sequence
  - 2.4 General Integration Information
- 3 Overview Client-Side Implementation
  - 3.1 Display Klarna Widget
  - 3.2 Klarna Authorize
- 4 Overview Server-Side API Calls
  - 4.1 Genericpayment start\_session
    - 4.1.1 Request "Generic Start Session"
    - 4.1.2 Response "Genericpayment - start\_session"
  - 4.2 Genericpayment update\_session
    - 4.2.1 Request "Genericpayment - update"
    - 4.2.2 Response "Genericpayment - update"
  - 4.3 Genericpayment cancel\_authorization
    - 4.3.1 Request "Genericpayment - cancel\_authorization"
    - 4.3.2 Response "Genericpayment - cancel\_authorization"
  - 4.4 Preauthorization / Authorization
    - 4.4.1 Request "preauthorization / authorization"
    - 4.4.2 Response "preauthorization / authorization"
      - 4.4.2.1 Common Parameter
      - 4.4.2.2 Parameter (APPROVED)
      - 4.4.2.3 Parameter (PENDING)
      - 4.4.2.4 Parameter (REDIRECT)
      - 4.4.2.5 Parameter (ERROR)
  - 4.5 Capture
    - 4.5.1 Request "capture"
    - 4.5.2 Response "Capture"
      - 4.5.2.1 Common Parameter
      - 4.5.2.2 Parameter (APPROVED)



Establishing Klarna Checkout is a 3 step process.

# 1

## Create a New Session

The first step is to setup a new checkout session by sending a genericpayment-request to our server API as listed below. This step is a pre-condition for loading the Klarna widget.

Please make sure to send the customer data within this request as it is needed to initiate a new session. A session will remain valid for 48 hours after the last update.

[Genericpayment start\\_session](#)

### Optional - Update the Session

If the customer leaves the checkout page and returns at a later point in time, you can update the created session and change for example the address data.

[Genericpayment update\\_session](#)

# 2

## Handle Token

Part of the response is a client token. This token needs to be passed to the Klarna Widget. This Widget sends the client token together with the selected payment method (e.g. "Klarna Pay Later") to Klarna to get a authorization token.

[Widget](#)

# 3

## Finalize the Checkout ((Pre-)Authorization)

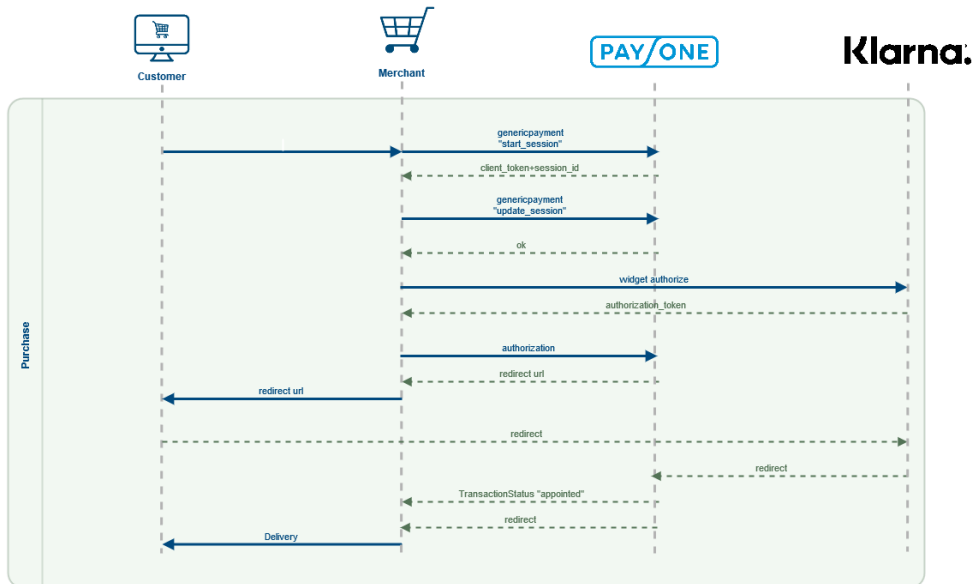
In the last step, you need to send the authorization token received by Klarna widget as part of the (pre-)authorization request to our server api.

[Preauthorization/Authorization](#)

The response contains a redirect-URL. You need to redirect the customer to this url, so that Klarna is able to securely handle data and optimize purchase flow. The customer doesn't need to take any action and will not notice this step in the background.

- 4.5.2.3 Parameter (ERROR)
- 4.6 Debit
  - 4.6.1 Request "Debit"
  - 4.6.2 Response "Debit"
    - 4.6.2.1 Common Parameter
    - 4.6.2.2 Parameter (APPROVED)
    - 4.6.2.3 Parameter (ERROR)
- 4.7 Refund
  - 4.7.1 Request "Refund"
  - 4.7.2 Response "Refund"
    - 4.7.2.1 Common Parameter
    - 4.7.2.2 Parameter (APPROVED)
    - 4.7.2.3 Parameter (ERROR)
- 4.8 Voucher Handling

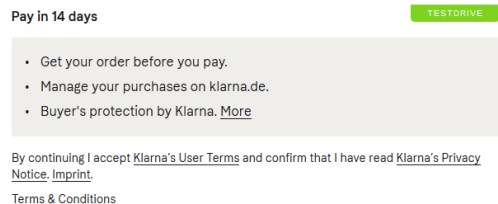
## Klarna Checkout Processing Sequence



## General Integration Information

There are a few things you need to take care of, when integrating Klarna checkout in your shop.

- You need to ensure, that the customer only triggers the session start by deliberately choosing Klarna as a payment method. As personal data will be sent to Klarna, the customer has to actively confirm this (for example: "By selecting this payment method we will send personal data to Klarna. Please confirm.").
- After confirmation of the payment method, you want to start with the first calls to server API as described above and then display the Klarna Widget. This Klarna Widget needs defined parameters to only display the chosen payment method, even if the widget is capable of displaying more than one Klarna payment method. For example after selecting "Klarna Pay Later", the widget should look something like this:



**Klarna.**

## Overview Client-Side Implementation

### Display Klarna Widget

The following integration steps are taken from Klarna documentation ( <https://developers.klarna.com/documentation/klarna-payments/integration-guide/present-klarna-widget> )

After creating a new session, you want to present the Klarna widget to your customer. This widget allows you to present all 3 payment methods offered by Klarna to the customer or define what payment method should be presented to the customer. The Klarna widget is rendered using JavaScript SDK.

1

Add SDK to your page (insert in body)

```
<script>
  window.klarnaAsyncCallback = function () {

    // This is where you start calling Klarna's JS SDK
    functions
    //
    // Klarna.Payments.init({...})

  };
</script>
<script src="https://x.klarnacdn.net/kp/lib/v1/api.js"
async></script>
```

2

Initialize SDK and place a container on your page. Please note that the Klarna widget requires a minimum width of 280px.

```
Klarna.Payments.init({
  client_token: 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJmb28iOiJiYXIIifQ.dtxWM6MIcgoeMgH87tGvsNDY6cH'
```

```
<div id="klarna_container"></div>
```

3

Load Klarna Widget

When loading the Klarna widget, one needs to provide the container ID established in the previous step and specify the `payment_method_category`. The `payment_method_category` specifies which of Klarna's customer offerings (e.g. Pay now, Pay later, or Slice it) will be shown in the widget.

```
Klarna.Payments.load({
  container: '#klarna-payments-container',
  payment_method_category: 'pay_later'
}, function (res) {
  console.debug(res);
})
```

4

Receive Response from load call

Klarna's widget uses the `show_form:true/false` field as a response flag to load and to authorize calls in the Javascript SDK. In addition the object "error" can be retrieved to receive further information about potential problems. The response information enables you to display to your customer Klarna as a payment method only when it can actually be utilised by your customer. In summary, there are three situations deriving from the "Receive Response" call.

**Positive Response**

If `show_form: true`, and there are no errors in the object returned, Klarna renders the payment options available to the customer in the widget.

```
{
  show_form: true
}
```

Adjust and Try Again

If `show_form: true`, but an `error` is returned as well, then something is wrong and the consumer needs to take action before moving forward. Klarna will inform the consumer about the details of the error in the widget. Optionally, you can interpret the invalid fields in the error message and take appropriate actions on your checkout page. See the [JavaScript SDK reference](#) page for further information.

```
{
  show_form: true,
  error: {
    invalid_fields: [
      billing_address.street_address
      billing_address.city
      billing_address.given_name
      billing_address.postal_code
      billing_address.family_name
      billing_address.email    ]
    }
  }
}
```

#### Negative Response

If `show_form: false`, the payment method chosen by the customer will not be offered for this order based on Klarna's evaluation. A message is displayed to the consumer in the Widget.

When Klarna returns a `show_form: false`, your store cannot offer the selected payment method to this customer. You need to present other payment methods to the customer and hide the option of Klarna payment.

```
{
  show_form: false
}
```

## Klarna Authorize

### 5

#### Authorise the Purchase

Once the customer has successfully selected Klarna as payment method in the checkout, the purchase can be authorised by using the Klarna JavaScript SDK. In this step all necessary customer details are transmitted to Klarna for taking a decision about whether or not to offer credit for the purchase. As a result of the authorisation you will receive a token enabling you to place an order towards Klarna. A successful authorisation guarantees that the order can be generated within 60 minutes.

#### Sending the Request

The Authorisation is initiated by a client-side call which will provide you with a token for the order once successfully completed.

Here is an example request for an authorisation call. In the call one must provide the billing and optionally the shipping address.

```

Klarna.Payments.authorize({
  payment_method_category: "pay_later"
}), {
  purchase_country: "GB",
  purchase_currency: "GBP",
  locale: "en-GB",
  billing_address: {
    given_name: "John",
    family_name: "Doe",
    email: "john@doe.com",
    title: "Mr",
    street_address: "13 New Burlington St",
    street_address2: "Apt 214",
    postal_code: "W13 3BG",
    city: "London",
    region: "",
    phone: "01895808221",
    country: "GB"
  },
  shipping_address: {
    given_name: "John",
    family_name: "Doe",
    email: "john@doe.com",
    title: "Mr",
    street_address: "13 New Burlington St",
    street_address2: "Apt 214",
    postal_code: "W13 3BG",
    city: "London",
    phone: "01895808221",
    country: "GB"
  },
  order_amount: 10,
  order_tax_amount: 0,
  order_lines: [{
    type: "physical",
    reference: "19-402",
    name: "Battery Power Pack",
    quantity: 1,
    unit_price: 10,
    tax_rate: 0,
    total_amount: 10,
    total_discount_amount: 0,
    total_tax_amount: 0,
    product_url: "https://www.estore.com/products/f2a8d7e34",
    image_url: "https://www.exampleobjects.com/logo.png"
  }],
  customer: {
    date_of_birth: "1970-01-01",
  },
}, function(res) {
  console.debug(res);
})

```

According to Klarna documentation ([Klarna Docs - Authorize the purchase](#)):

#### User interaction during the authorize call

When authorizing the order, Klarna conducts a full risk assessment. Therefore, from the point where you call authorize until you receive the callback you must:

1. Avoid sending another authorize call (e.g. disable the buy button from being clicked again)
2. Show to the consumer that the order is being processed (e.g. by showing a loading spinner)
3. Prevent consumer from changing order or billing details (e.g. lock the input fields on your page)

The callback is typically received within seconds, but may take up to a minute or so in case a consumer sign-up is required when the user interacts with the widget.

## 6

### Process callback from authorise call

Once the widget has processed the authorisation a callback is performed providing an object with one of the following parameters:

- `approved (true/false)` - the authorization result, approved or denied
- `show_form (true/false)` - whether the Klarna Widget should be displayed or hidden
- `authorization_token` - a token which allows you to place the order via a server side call, only returned if the authorization was approved
- `error` - contains details of potential error messages

#### Order approved

If `approved` carries the value "true", Klarna has approved the authorisation of credit for this order. With the `authorization_token` the order can be completed using a server side place order call. The token is valid for 60 minutes during which the authorisation is guaranteed.

```
{
  authorization_token: "b4bd3423-24e3",
  approved: true,
  show_form: true
}
```

#### Order not approved

If `approved: false`, Klarna cannot approve the purchase. There are now two options:

##### Option 1 - Fixable error

In the case of an adjustable error, you will receive `show_form: true` and a specification about what fields that are invalid, e.g. `error: { invalid_fields: ["billing_address.email"] }`. The widget will also display an error message to the consumer, asking them to correct it before you re-authorize the order. This error message will also clarify which specific field that is incorrect.

```
{
  approved: false,
  show_form: true,
  error: {
    invalid_fields: [
      billing_address.street_address
      billing_address.city
      billing_address.given_name
      billing_address.postal_code
      billing_address.family_name
      billing_address.email ]
  }
}
```

##### Option 2 - Order declined

If `show_form: false`, the order is declined and the Klarna widget should be hidden requesting the customer to select a different method of payment.

```
{
  approved: false,
  show_form: false
}
```

## 7

### Finalize the authorisation

According to Klarna documentation ([Klarna Docs - Authorize the purchase](#)) the following steps on client side are optional.

Finalising an authorisation only applies to your integration if you offer payment methods where funds are drawn from the consumers directly, such as bank transfer or Sofort, in a multi-step checkout. If you integrate a multi-step checkout, you may call `authorize()` with the `auto_finalize: false` property set in order to indicate that there is a finalization step. In this case the response may differ.

In a multi-step checkout scenario the `authorize()` call can be triggered when the consumer selects the payment method and then presses the "continue" button to go to the next step of the checkout. With Pay Now as payment method category however transferring the funds should only happen once the consumer has pressed the "buy" button to finalize the purchase.

To cater for such a scenario `authorize()` can still be called when the consumer has selected the payment method, but with the `auto_finalize` property set to `false`. `authorize()` example:

```
Klarna.Payments.authorize(  
  { payment_method_category: 'pay_now', auto_finalize: false },  
  {},  
  function(res) {  
    // proceed to next checkout page. The finalize_required  
    // property in the response indicates  
    // if finalize is needed or not.  
    //  
    // res = {  
    //   show_form: true,  
    //   approved: false,  
    //   finalize_required: true  
    // }  
  })
```

Now when the consumer reached the last page in the checkout and can finalize the purchase `finalize()` is called. This will then trigger the transfer of funds and return the authorization token in the finalize callback. The flow is transparent to all payment method categories. That means if finalization was not needed in the `authorize()` call (e.g. for pay later) `finalize()` can still be called and will return the `authorization_token` so that the implementation remains the same for all payment method categories. `finalize()` example:

```
Klarna.Payments.finalize(  
  { payment_method_category: 'pay_now' },  
  {},  
  function(res) {  
    // res = {  
    //   show_form: true,  
    //   approved: true,  
    //   authorization_token: ...  
    // }  
  })
```

---

## Overview Server-Side API Calls

### Genericpayment start\_session

The first step is to setup a new checkout session by sending a genericpayment-request to our server API as listed below. This step is a pre-condition for loading the Klarna widget.

Please make sure to send the customer data within this request as it is needed to initiate a new session. A session will remain valid for 48 hours after the last update.

Only submit a buyer's personal information within this request if you ensure the buyer has actively chosen Klarna as a payment method - due to GDPR.

---



## Request "Genericpayment - start\_session"

Request "Generic Start Session"		
API parameter	Required	Comments
<a href="#">request</a>	+	Fixed Value <code>genericpayment</code>
<code>add_paydata</code> <code>[action]</code>	+	identifies the request as one to start a new session  Fixed Value <code>start_session</code>
<code>add_paydata</code> <code>[merchant_data]</code>	-	"EMD" - Extra Merchant Data - means that integration with any such package depends on  (i) merchant's offering and  (ii) merchant's technical possibilities.  It is not to be interpreted in such a way that EMD is not required at all.  Format <code>BASE 64 String</code>
<a href="#">amount</a>	+	Specifies the total gross amount of a payment transaction.  Value is given in smallest currency unit, e.g. Cent of Euro; Pence of Pound sterling; Öre of Swedish krona.  The amount must be less than or equal to the amount of the corresponding booking.  Format <code>NUMERIC(1..10)</code> Permitted values  max. +/- 19 999 999 99
<a href="#">currency</a>	+	Specifies currency for this transaction  Format <code>LIST</code> Permitted values <code>ISO 4217 (currencies) 3-letter-codes</code> Samples  EUR allowed Values <code>EUR</code>  USD                      CHF  GBP                      DKK  SEK  NOK
<a href="#">country</a>	+	Specifies country of address for the customer Format <code>LIST</code> Permitted values <code>ISO 3166 2-letter-codes</code> Samples <code>DE</code>  GB US Some cou  Countries require additional information in parameter " <a href="#">state</a> "

firstname	-	<p>First name of customer; optional if company is used, i.e.: you may use</p> <ul style="list-style-type: none"> <li>• "company"</li> <li>• or "lastname"</li> <li>• or "firstname" plus "lastname"</li> </ul> <p>FormatCHAR(1..50)</p>
lastname	-	<p>Last name of customer; optional if company is used, i.e.: you may use</p> <ul style="list-style-type: none"> <li>• "company"</li> <li>• or "lastname"</li> <li>• or "firstname" plus "lastname"</li> </ul> <p>FormatCHAR(2..50)</p>
company	-	<p>Company name of customer; The company name is optional if lastname is used, i.e.: you may use</p> <ul style="list-style-type: none"> <li>• "company"</li> <li>• or "lastname"</li> <li>• or "firstname" plus "lastname"</li> </ul> <p>FormatCHAR(2..50)</p> <p>Submitting this parameter leads to a B2B transaction. Availability of B2B transactions is limited to some countries.</p>
title	-	<p>Title of the customer FormatCHAR(1..20) SamplesDr Prof. Dr. -Ing.</p>
street	-	<p>Street number and name (required: at least one character) FormatCHAR(1..50)</p>
zip	-	<p>Postcode FormatCHAR(2..10) Permitted Symbols[0-9][A-Z][a-z][ _.-/ ]</p>
city	-	<p>City of customer FormatCHAR(2..50)</p>
state	-	<p>Specifies state / region of country for the customer. "state" is required for these countries: US, CA, CN, JP, MX, BR, AR, ID, TH, IN <b>and must not be used for all other countries.</b> FormatLIST Permitted valuesISO 3166-2 States (regions) 2- Samples USAK letter-codes AL AR Samples CAAB BC</p>

addressaddition	-	Specifies an additional address line for the invoice address of the customer. FormatCHAR(1..50) Samples7th floor c/o Maier
gender	-	Gender of customer (female / male / diverse* ) FormatLIST Permitted valuesf * currently not in use m d
ip	-	Customer's IP-V4-address (123.123.123.123) or IP-V6-address FormatCHAR(1..39)
email	-	email-address of customer FormatCHAR(5..254) Permitted SymbolsRFC 5322
telephonenumber	-	Phone number of customer FormatCHAR(1..30)
birthday	-	Date of birth of customer  FormatDATE(8), YYYYMMDD Samples20190101 19991231 mandatory for Austria, Germany, Switzerland and the Netherlands.
language	-	Language indicator (ISO 639) to specify the language that should be presented to the customer (e.g. for error messages, frontend display).  If the language is not transferred, the browser language will be used. For a non-supported language English will be used.  FormatLIST Permitted valuesISO 639-1 (Language) 2-letter-codes
shipping_firstname	-	First name of delivery address FormatCHAR(1..50)
shipping_lastname	-	Surname of delivery address FormatCHAR(1..50)
add_paydata [shipping_title]	-	Title recipient within the delivery address  FormatCHAR(1..50)
shipping_company	-	Company Name of the delivery address FormatCHAR(2..50)
shipping_street	-	Street number and name of delivery address FormatCHAR(2..50)

shipping_zip	-	<p>Postcode of delivery address  FormatCHAR(2..10) Permitted Symbols  [0-9][A-Z][a-z][_./ ]</p>
shipping_city	-	<p>City of delivery address  FormatCHAR(2..50)</p>
shipping_country	-	<p>Specifies country of delivery address for the customer  FormatLIST Permitted valuesISO 3166 2- letter-codes SamplesDE  GBSo  me  US  cou  ntri  es require additional information in parameter <a href="#">shipping_state</a></p>
shipping_state	-	<p>Specifies state of country of delivery address for the customer  "shipping_state" is required for these countries: US, CA, CN, JP, MX, BR, AR, ID, TH, IN (if shipping_country is given) <b>and must not be used in all other countries.</b>  FormatLIST  Permitted valuesISO 3166-2 States (regions) 2- letter-codes Samples USAK  AL  AR  Samples CAAB  BC</p>
successurl	+	<p><b>successurl - definition</b></p> <p>URL for "payment successful"  FormatCHAR(2..255) Scheme  &lt;scheme&gt;://&lt;host&gt;/&lt;path&gt;  &lt;scheme&gt;://&lt;host&gt;/&lt;path&gt;?&lt;query&gt;</p> <p>scheme-pattern: [a-zA-Z]{1}[a-zA-Z0-9]{1,9}</p>
errorurl	+	<p><b>errorurl - definition</b></p> <p>URL for "faulty payment"  FormatCHAR(2..255) Scheme  &lt;scheme&gt;://&lt;host&gt;/&lt;path&gt;  &lt;scheme&gt;://&lt;host&gt;/&lt;path&gt;?&lt;query&gt;</p> <p>scheme-pattern: [a-zA-Z]{1}[a-zA-Z0-9]{1,9}</p>

backurl	+	<p><b>backurl - definition</b></p> <p>URL for "Back" or "Cancel"  FormatCHAR(2..255)Scheme</p> <p>&lt;scheme&gt;://&lt;host&gt;/&lt;path&gt;</p> <p>&lt;scheme&gt;://&lt;host&gt;/&lt;path&gt;?&lt;query&gt;</p> <p>scheme-pattern: [a-zA-Z]{1}[a-zA-Z0-9]{1,9}</p>
add_paydata [shipping_telephonenumber]	-	<p>telephone number of the recipient or a contact at the delivery address</p> <p>FormatCHAR(1..30)</p>
add_paydata [shipping_email]	-	<p>email-address of the recipient or a contact at the delivery address</p> <p>FormatCHAR(5..254)Permitted SymbolsRFC 5322</p> <p>This parameter is required if a shipping address is given.</p>
add_paydata [last_four_ssn]	-	<p>Last four digits for customer social security number.</p> <p>FormatNUMERIC(4)</p>
add_paydata [organization_entity_type]	-	<p>Only relevant for B2B transactions.</p> <p>Permitted ValuesLIMITED_COMPANY  PUBLIC_LIMITED_COMPANY  ENTREPRENEURIAL_COMPANY  LIMITED_PARTNERSHIP_LIMITED_COMPANY  LIMITED_PARTNERSHIP  GENERAL_PARTNERSHIP  REGISTERED_SOLE_TRADER  SOLE_TRADER  CIVIL_LAW_PARTNERSHIP  PUBLIC_INSTITUTION  OTHER</p>
add_paydata [organization_registry_id]	-	<p>will be given to Klarna as "vat_id"</p> <p>FormatCHAR(1..50)</p>
personalid	-	<p>Person specific numbers or characters, e.g. number of passport / ID card</p> <p>FormatCHAR(1..32)Permitted Symbols[0-9][A-Z][a-z][+-. / ( )]</p> <p>Mandatory for Denmark, Finland, Norway and Sweden</p>
it[n]	+	<p>Klarna item type</p> <p>Permitted Valuesgoods : Goods  shipment : Shipping charges  handling : Handling fee  voucher : Voucher / discount</p>
id[n]	+	<p>Product number, SKU, etc. of this item</p> <p>FormatCHAR(1..32)Array</p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p> <p>Permitted Symbols[0-9][A-Z][a-z][() [] {} +_#/:]</p>

pr[n]	+	<p>Unit gross price of the item in smallest unit! e.g. cent</p> <p>FormatNUMERIC(10) max. 19 999 999 99Array</p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p>
no[n]	+	<p>Quantity of this item</p> <p>FormatNUMERIC(6) Array</p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p>
de[n]	+	<p>Description of this item. Will be printed on documents to customer.</p> <p>FormatCHAR(1..255) Array</p> <p><b>Example</b></p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p> <pre> de[1]=Product 1 de[2]=Product 2 de[3]=Product 3 ... de[400]=Product 400 </pre>
va[n]	+	<p>VAT rate (% or bp)</p> <p>FormatNUMERIC(4) Array</p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p>

#### Genericpayment start\_session - Example Request

```

mid=23456 (your mid)
portalid=12345123 (your portalid)
key=abcdefghijklmn123456789 (your key)
api_version=3.10
mode=test (set to „live“ for live-requests)
request=genericpayment
encoding=UTF-8
aid=12345 (your aid)
clearingtype=fnc
financingtype=KIS
amount=3000
currency=EUR
lastname=Approved
firstname=Testperson-de
salutation=Herr
country=DE
language=de
gender=m
add_paydata[action]=start_session
birthday=19600707
street=Hellersbergstraße 14
city=Musterstadt
zip=12345
email=youreemail@email.com
telephonenumber=01512345678
de[1]=for the feet
id[1]=socks1
it[1]=goods
no[1]=1
pr[1]=2340
va[1]=1900

```

Response "Genericpayment - start\_session"

Response "Genericpayment - start_session"		
API parameter	Required	Comments
add_paydata [session_id]	+	Identifier for the started session at Klarna  Sample068df369-13a7-4d47-a564-62f8408bb760
add_paydata [client_token]	+	Client token to authorize the session for payment via Klarna Widget
add_paydata [authorized_payment_method]	+	Name of payment that got authorized for customer payment. (You will only receive the parameters according to the payment method you selected via clearingtype and financingtype)
add_paydata [payment_method_category_name]	+	Name of Klarna payment category  Payment Method Category to show on the Payment Page.  Permitted ValuesDIRECT_DEBIT  DIRECT_BANK_TRANSFER  PAY_NOW  PAY_LATER  PAY_OVER_TIME
add_paydata [payment_method_category_identifier]	+	Identifier for Klarna payment category  Permitted ValuesPay_later  Pay_now  Pay_over_time  Direct_bank_transfer  Direct_debit
add_paydata [payment_method_category_asset_url_descriptive]	+	URL of Klarna payment category assets for descriptive design (payment method identified by small icon and test on badge)  Sample https://x.klarnacdn.net/payment-method/assets/badges/generic/klarna.svg
add_paydata [payment_method_category_asset_url_standard]	+	URL of Klarna payment category assets for standard design (payment method only identified by small icon on badge)  Sample https://x.klarnacdn.net/payment-method/assets/badges/generic/klarna.svg
status	+	Permitted ValuesAPPROVED  ERROR
<b>APPROVED</b>		
workorderid	+	The workorderid is a technical id returned from the PAYONE platform to identify a workorder. A workorder is a part of a payment process (identified by a txid). The workorderid is used for the genericpayment request. FormatCHAR(1..50)
<b>ERROR</b>		

errorcode	+	In case of error the PAYONE Platform returns an error code for your internal usage. FormatNUMERIC(1..6)
errormessage	+	In case of error the PAYONE Platform returns an error message for your internal usage. FormatCHAR(1..1024)
customermessage	+	The customermessage is returned to your system in order to be displayed to the customer.  (Language selection is based on the end customer's language, parameter "language") FormatCHAR(1..1024)

### Genericpayment start\_session - Example Response

```
status=OK
add_paydata[payment_method_category_asset_url_descriptive]=https://x.
klarnacdn.net/payment-method/assets/badges/generic/klarna.svg
add_paydata[client_token]=eyJhbGciOiJSUzI1NiJ9.
ewogICJzZXNzaW9uX2lkIiA6ICI0N2EzNWlWOC1kNWVlLTcxYjU0OTVhZmM1YVU3NTFjNWJlZmI
iLAogICJiYXNlX3VybyCIGoiAiaHR0cHM6Ly9rbGFybmEtcGF5bWVudHMtZXUucGxheWdyb3VuZC
5rbGFybmEuY29tIiwKIiCAiZGVzaWduIiA6ICJrbGFybmEiLAogICJzYW5ndWFnZSIgOiAiZW4iL
AogICJwdXJjaGFzZV9jb3VudHJ5IiA6ICJERSIsCiAgInRyYWNlX2Zsb3ciIDogZmFsc2UsCiAg
ImVudml5b25tZW50IiA6ICJwbGF5Z3JvdW5kIiwKIiCAibWVyY2hhbnRfbmFtZSIgOiAiWW91ciB
idXNpbmVzcyBuYW11IiwKIiCAic2Vzc2lvd190eXB1IiA6ICJQVlNRU5UUyIsCiAgImNsaWVudF
9ldmVudF9iYXNlX3VybyCIGoiAiaHR0cHM6Ly9ldnQucGxheWdyb3VuZC5rbGFybmEuY29tIiwKI
CAiZXhwZXJpbWVudHMiIDogWyBdCn0.
gD6XVLaZeL541T1QYkGsZ01wbAYuTYyhIXcR8irG461U9w1HLugSji4_jc8LsE7cVMMVqLph_CJ
zCAiZNKzsc9AW6Mf88X8VdR9gV7owv3EVqEdK2K9-g-
26Wu2pGE88Um2z2Iz7uUUzbhn6d4rI4CICOLsdaVXJ7MKhzPgnW-a6DM-
K9JP2RLIfLJKjn3wCjEtJSc_8gqj3_fv8YHaTWRkBD9pUGCBB0rUS-
komFAPBdzZiCABiw5LsdK9dWpI2wszKeE_iPd6NuxGr4U37XtIX9Gc-
uTfn21o0JUJddbiPtItmetWQSY_mru004WlVQxuzvXk7XYwCCb8rrXYw
add_paydata[payment_method_category_identifier]=pay_over_time
add_paydata[session_id]=47a35b08-d5ee-71b5-95d1-eae751c5befb
add_paydata[payment_method_category_name]=Slice it.
add_paydata[payment_method_category_asset_url_standard]=https://x.
klarnacdn.net/payment-method/assets/badges/generic/klarna.svg
workorderid=WX1A37YBGD9D11DK
```

## Genericpayment update\_session

In order to update the session, e.g. the customer has added a new item to the cart, you need to send a genericpayment-request with action=update. The call needs to provide the new list of items representing the complete shopping cart.

- An update is only possible as long as the preauthorization is not captured completely.
- Do not send the difference/changes, instead you need to send the **complete** new item list
- The amount cannot be higher than the amount of the preauthorization. A lower amount is allowed.

## Request "Genericpayment - update"



## Request "Genericpayment - update"

API parameter	Required	Comments
<a href="#">request</a>	+	Fixed Value <code>genericpayment</code>
<code>add_paydata</code> [action]	+	Fixed value <code>update_session</code>
<code>add_paydata</code> [reservation_txid]	+	Referencing the reservation (received from <code>authorization.response</code> -> <code>add_paydata[reservation_txid]</code> )
<code>add_paydata</code> [session_id]	+	Identifier for the started session at Klarna  Sample <code>068df369-13a7-4d47-a564-62f8408bb760</code>
<a href="#">workorderid</a>	+	Referencing the reservation (received from <code>authorization.response</code> -> <code>add_paydata[workorderid]</code> )  The <code>workorderid</code> is a technical id returned from the PAYONE platform to identify a workorder. A workorder is a part of a payment process (identified by a <code>txid</code> ). The <code>workorderid</code> is used for the <code>genericpayment</code> request. Format <code>CHAR(1..50)</code>
<a href="#">country</a>	+	Specifies country of address for the customer Format <code>LIST</code> Permitted values <code>ISO 3166 2-letter-codes</code> Samples <code>DE</code>  <code>GB</code> <code>US</code> Some countries require additional information in parameter " <a href="#">state</a> "
<a href="#">amount</a>	+	Specifies the total gross amount of a payment transaction.  Value is given in smallest currency unit, e.g. Cent of Euro; Pence of Pound sterling; Öre of Swedish krona.  The amount must be less than or equal to the amount of the corresponding booking.  Format <code>NUMERIC(1..10)</code> Permitted values <code>max. +/- 19 999 999 99</code>
<a href="#">currency</a>	+	Specifies currency for this transaction  Format <code>LIST</code> Permitted values <code>ISO 4217 (currencies) 3-letter-codes</code> Samples <code>EUR</code>  <code>USD</code>  <code>GBP</code>
<a href="#">financingtype</a>	+	Permitted Values <code>KIS</code> <code>KIV</code> <code>KDD</code>

it[n]	+	<p>Klarna item type</p> <p>Permitted Values</p> <pre>goods      : Goods shipment   : Shipping charges handling   : Handling fee voucher    : Voucher / discount</pre>
id[n]	+	<p>Product number, SKU, etc. of this item</p> <p>Format CHAR(1..32) Array</p> <p>AN..32</p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p> <p>Permitted Symbols [0-9][A-Z][a-z][(){}+_-#/:]</p>
pr[n]	+	<p>Unit gross price of the item in smallest unit! e.g. cent</p> <p>Format NUMERIC(10) max. 19 999 999 99 Array</p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p>
no[n]	+	<p>Quantity of this item</p> <p>Format NUMERIC(6) Array</p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p>
de[n]	+	<p>Description of this item. Will be printed on documents to customer.</p> <p>Format CHAR(1..255) Array</p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p><b>Example</b></p> <pre>de[1]=Product 1 de[2]=Product 2 de[3]=Product 3 ... de[400]=Product 400</pre> </div> <p>AN..50</p>
va[n]	+	<p>VAT rate (% or bp)</p> <p>Format NUMERIC(4) Array</p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p>
narrative_text	-	<p>Dynamic text element on account statements</p> <p>Format CHAR(1..81)</p> <p>(3 lines with 27 characters each) and credit card statements.</p>

## Response "Genericpayment - update"

Response "Genericpayment - update"		
API parameter	Required	Comments
status	+	<p>Permitted Values</p> <pre>APPROVED ERROR</pre>

APPROVED		
workorderid	+	The workorderid is a technical id returned from the PAYONE platform to identify a workorder. A workorder is a part of a payment process (identified by a txid). The workorderid is used for the genericpayment request. FormatCHAR(1..50)
ERROR		
errorcode	+	In case of error the PAYONE Platform returns an error code for your internal usage. FormatNUMERIC(1..6)
errormessage	+	In case of error the PAYONE Platform returns an error message for your internal usage. FormatCHAR(1..1024)
customermessag e	+	The customermessag e is returned to your system in order to be displayed to the customer.  (Language selection is based on the end customer's language, parameter "language") FormatCHAR(1..1024)
addressedat a [reservation_tx id]		The reservation_txid is a technical id returned from the PAYONE platform to identify a reservation. A reservation is a part of a payment process (identified by a txid). The reservation is used for "genericrequest". FormatCHAR(12..50) Permitted Symbols[0-9,A-Z]
addressedat a[workorderid]		The workorderid is a technical id returned from the PAYONE platform to identify a workorder. A workorder is a part of a payment process (identified by a txid). The workorderid is used for the genericpayment request. FormatCHAR(1..50)
addressedat a [authorized_pay ment_method]		Name of payment that got authorized for customer payment. (You will only receive the parameters according to the payment method you selected via clearingtype and financingtype)

## Genericpayment cancel\_authorization

To update an order, you need to send a genericpayment-request with action=update. The call needs to provide the new list of items representing the complete shopping cart.

- An update is only possible as long as the preauthorization is not captured completely.
- Don't send the difference/changes, instead you need to send the **complete** new item list
- The amount can't be higher as the amount of the preauthorization. A lower amount is allowed.

## Request "Genericpayment - cancel\_authorization"

**Request "Genericpayment - cancel\_authorization"**

API parameter	Required	Comments
request	+	Fixed Valuegenericpayment
add_paydata[action]	+	cancel_authorization
add_paydata [authorization_token]	+	Token of the authorization, which is to be canceled. Will be provided by Klarna's JS API

## Response "Genericpayment - cancel\_authorization"

Response "Genericpayment - cancel_authorization"		
API parameter	Required	Comments
status	+	Permitted ValuesAPPROVED ERROR
<b>APPROVED</b>		
workorderid	+	The workorderid is a technical id returned from the PAYONE platform to identify a workorder. A workorder is a part of a payment process (identified by a txid). The workorderid is used for the genericpayment request. FormatCHAR ( 1 . . 50 )
<b>ERROR</b>		
errorcode	+	In case of error the PAYONE Platform returns an error code for your internal usage. FormatNUMERIC ( 1 . . 6 )
errormessage	+	In case of error the PAYONE Platform returns an error message for your internal usage. FormatCHAR ( 1 . . 1024 )
customermessage	+	The customermessage is returned to your system in order to be displayed to the customer.  (Language selection is based on the end customer's language, parameter "language") FormatCHAR ( 1 . . 1024 )

## Preauthorization / Authorization

### Request "preauthorization / authorization"

Request "preauthorization / authorization"		
API parameter	Required	Comments

add_paydata [authorization_token]	+	Sample987d6543-f27b-45t5-b378-245465456780FormatString
firstname	+	<p>First name of customer; optional if company is used, i.e.: you may use</p> <ul style="list-style-type: none"> <li>• "company"</li> <li>• or "lastname"</li> <li>• or "firstname" plus "lastname"</li> </ul> <p>FormatCHAR(1..50) full address required and must match the shipping-address</p>
lastname	+	<p>Last name of customer; optional if company is used, i.e.: you may use</p> <ul style="list-style-type: none"> <li>• "company"</li> <li>• or "lastname"</li> <li>• or "firstname" plus "lastname"</li> </ul> <p>FormatCHAR(2..50) full address required and must match the shipping-address</p>
company	-	<p>If filled, the transaction is marked as B2B.</p> <p>Company name of customer; The company name is optional if lastname is used, i.e.: you may use</p> <ul style="list-style-type: none"> <li>• "company"</li> <li>• or "lastname"</li> <li>• or "firstname" plus "lastname"</li> </ul> <p>FormatCHAR(2..50)</p>
street	+	<p>Street number and name (required: at least one character)</p> <p>FormatCHAR(1..50) full address required and must match the shipping-address</p>
zip	+	<p>Postcode</p> <p>FormatCHAR(2..10) Permitted Symbols[0-9][A-Z][a-z][ _.-/ ] full address address</p> <p>required and must match the shipping-address</p>
city	+	<p>City of customer</p> <p>FormatCHAR(2..50) full address required and must match the shipping-address</p>
country	+	<p>Specifies country of address for the customer</p> <p>FormatLIST Permitted valuesISO 3166 2-letter-codes SamplesDE GB US Some countries S</p> <p>require additional information in parameter "<a href="#">state</a>"</p> <p>full address required and must match the shipping-address</p>
addressaddition	o	<p>Specifies an additional address line for the invoice address of the customer.</p> <p>FormatCHAR(1..50) Samples7th floor mandatory for NL c/o Maier</p>

gender	o	<p>Gender of customer (female / male / diverse* )</p> <p>FormatLIST Permitted valuesf</p> <p>m * currently not in use</p> <p>d mandatory for Austria, Germany and the Netherlands</p> <p>"d" is currently not supported by Klarna</p>
ip	+	<p>Customer's IP-V4-address (123.123.123.123) or IP-V6-address</p> <p>FormatCHAR(1..39)</p>
email	+	<p>email-address of customer</p> <p>FormatCHAR(5..254) Permitted SymbolsRFC 5322</p>
telephonenumber	-	<p>Phone number of customer</p> <p>FormatCHAR(1..30)</p>
birthday	o	<p>Date of birth of customer</p> <p>FormatDATE(8), YYYYMMDDSamples20190101</p> <p>19991231 mandatory for Austria, Germany, Switzerland and the Netherlands.</p>
shipping_firstname	-	<p>First name of delivery address</p> <p>FormatCHAR(1..50)</p>
shipping_lastname	-	<p>Surname of delivery address</p> <p>FormatCHAR(1..50)</p>
shipping_company	-	<p>Company Name of the delivery address</p> <p>FormatCHAR(2..50)</p>
shipping_street	-	<p>Street number and name of delivery address</p> <p>FormatCHAR(2..50)</p>
shipping_zip	-	<p>Postcode of delivery address</p> <p>FormatCHAR(2..10) Permitted Symbols[0-9][A-Z][a-z][_./ ]</p>
shipping_city	-	<p>City of delivery address</p> <p>FormatCHAR(2..50)</p>
shipping_country	-	<p>Specifies country of delivery address for the customer</p> <p>FormatLIST Permitted valuesISO 3166 2-letter-codes SamplesDE</p> <p>GBSome countries require additional information in parameter <a href="#">shipping_state</a></p>

successurl	+	<p><b>successurl - definition</b></p> <p>URL for "payment successful"  FormatCHAR(2..255)Scheme</p> <p>&lt;scheme&gt;://&lt;host&gt;/&lt;path&gt;</p> <p>&lt;scheme&gt;://&lt;host&gt;/&lt;path&gt;?&lt;query&gt;</p> <p>scheme-pattern: [a-zA-Z]{1}[a-zA-Z0-9]{1,9}</p>
errorurl	+	<p><b>errorurl - definition</b></p> <p>URL for "faulty payment"  FormatCHAR(2..255)Scheme</p> <p>&lt;scheme&gt;://&lt;host&gt;/&lt;path&gt;</p> <p>&lt;scheme&gt;://&lt;host&gt;/&lt;path&gt;?&lt;query&gt;</p> <p>scheme-pattern: [a-zA-Z]{1}[a-zA-Z0-9]{1,9}</p>
backurl	+	<p><b>backurl - definition</b></p> <p>URL for "Back" or "Cancel"  FormatCHAR(2..255)Scheme</p> <p>&lt;scheme&gt;://&lt;host&gt;/&lt;path&gt;</p> <p>&lt;scheme&gt;://&lt;host&gt;/&lt;path&gt;?&lt;query&gt;</p> <p>scheme-pattern: [a-zA-Z]{1}[a-zA-Z0-9]{1,9}</p>
personalid	0	<p>Person specific numbers or characters, e.g. number of passport / ID card  FormatCHAR(1..32)Permitted Symbols[0-9][A-Z][a-z][+-. /()] Mandator  y for</p> <p>Denmark, Finland, Norway and Sweden</p>
it[n]	+	<p>Klarna item type</p> <p>Permitted Valuesgoods : Goods</p> <p>shipment : Shipping charges</p> <p>handling : Handling fee</p> <p>voucher : Voucher / discount</p>
id[n]	+	<p>Product number, SKU, etc. of this item  FormatCHAR(1..32)Array</p> <p>Array elements [n] starting with [1]; serially numbered; max [400]  Permitted Symbols[0-9][A-Z][a-z][(){}+_-#/:]</p>

pr[n]	+	<p>Unit gross price of the item in smallest unit! e.g. cent</p> <p>FormatNUMERIC(10) max. 19 999 999 99Array</p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p>
no[n]	+	<p>Quantity of this item</p> <p>FormatNUMERIC(6) Array</p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p>
de[n]	+	<p>Description of this item. Will be printed on documents to customer.</p> <p>FormatCHAR(1..255) Array</p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p> <div data-bbox="423 480 1066 722" style="border: 1px solid gray; padding: 5px;"> <p><b>Example</b></p> <pre> de[1]=Product 1 de[2]=Product 2 de[3]=Product 3 ... de[400]=Product 400 </pre> </div>
va[n]	+	<p>VAT rate (% or bp)</p> <p>FormatNUMERIC(4) Array</p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p>
workorderid	+	<p>The workorderid is a technical id returned from the PAYONE platform to identify a workorder. A workorder is a part of a payment process (identified by a txid). The workorderid is used for the genericpayment request.</p> <p>FormatCHAR(1..50)</p>



### Authorization - Example Request

```
add_paydata[authorization_token]=aaa-bbbb-11111-2222-ccc
aid=12346
amount=8540
api_version=3.10
backurl=https://meine.test.url.de/Checkout/back
birthday=19600707
city=Neuss
clearingtype=fnc
country=DE
currency=EUR
de[1]=for ocean use only
email=klarna@approved.de
encoding=UTF-8
errorurl=https://meine.test.url.de/Checkout/error
financingtype=KIS
firstname=Testperson-de
gender=m
id[1]=boat66
it[1]=goods
key=e123dc456a
language=de
lastname=Approved
mid=12345
mode=test
narrative_text=0123456789012345678901234567890123456789
no[1]=1
portalid=2000123
pr[1]=8540
reference=R15595689952007
request=authorization
salutation=Herr
street=Hellensbergstraße 14
successurl=https://meine.test.url.de/Checkout/success
telephonenumber=01522113356
va[1]=1900
zip=41460
```

### Response "preauthorization / authorization"

Response "preauthorization / authorization"			
Common Parameter	Required	Format	Comment
status	+		Permitted Values REDIRECT ERROR PENDING
add_paydata [workorderid]	-		Klarna Invoice: Workorder ID, used for update calls (cart updates)
add_paydata [reservation_txid]	-		Klarna Invoice: Reservation ID, used for update calls (cart updates)
Parameter (APPROVED)			
txid	+		The txid specifies the payment process within the PAYONE platform Format NUMERIC(9..12)

userid	+		PAYONE User ID, defined by PAYONE FormatNUMERIC(6..12)
<b>Parameter (PENDING)</b>			
txid	+		The txid specifies the payment process within the PAYONE platform FormatNUMERIC(9..12)
userid	+		PAYONE User ID, defined by PAYONE FormatNUMERIC(6..12)
<b>Parameter (REDIRECT)</b>			
txid	+		The txid specifies the payment process within the PAYONE platform FormatNUMERIC(9..12)
userid	+		PAYONE User ID, defined by PAYONE FormatNUMERIC(6..12)
redirecturl	+		Redirect URL. zMerchant system has to redirect customer to this URL to complete payment FormatCHAR(2..2000)
<b>Parameter (ERROR)</b>			
errormessage	+		In case of error the PAYONE Platform returns an error message for your internal usage. FormatCHAR(1..1024)
customermesssage	-		The customermesssage is returned to your system in order to be displayed to the customer.  (Language selection is based on the end customer's language, parameter " <a href="#">language</a> ") FormatCHAR(1..1024)

#### Authorization - Example Response

```

redirecturl=https://klarna-payments-eu.playground.klarna.com/v1/sessions
/aaa-bbbb-11111-2222-ccc/redirect
status=REDIRECT
txid=30343747
userid=7890123

```

## Capture

In the capturing step either the full amount or a partial amount can be captured. It is also possible to split the capturing in two steps, capturing an initial amount and subsequently capturing the remaining outstanding amount. Please note that the amount in the capture request must be equal or less than the remaining authorised amount.

In the capture call one can provide the order lines that are part of the shipment. By using the exact same order line name and reference as in the order, you will improve the customer's experience in the Klarna App.

## Request "capture"

Request "capture"														
API parameter	Required	Comments												
txid	+	<p>The txid specifies the payment process within the PAYONE platform FormatNUMERIC(9..12)</p>												
sequencenumber	0	<p>Sequence number for this transaction within the payment process (1..n), e.g. PreAuthorization: 0, 1. Capture: 1, 2. Capture: 2</p> <p>Required for multi partial capture (starting with the 2<sup>nd</sup> capture) FormatNUMERIC(1..3) Permitted values0..127</p>												
capturemode	+	<p>Parameter capturemode is mandatory to indicate whether this capture will be the last one. (Default: completed)</p> <p>Specifies whether this capture is the last one or whether there will be another one in future. FormatLIST</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Comment</th> </tr> </thead> <tbody> <tr> <td>completed</td> <td> <ul style="list-style-type: none"> <li>Set with last capture; i.e.: Delivery completed.</li> <li>No further capture is allowed.</li> </ul> </td> </tr> <tr> <td>notcompleted</td> <td> <ul style="list-style-type: none"> <li>Set with partial deliveries (last delivery with "completed")</li> <li>Another capture is expected to complete the transaction.</li> </ul> </td> </tr> </tbody> </table> <p>Payment specific:</p> <table border="1"> <thead> <tr> <th>Payment type</th> <th>Comment</th> </tr> </thead> <tbody> <tr> <td>PDT</td> <td>Parameter "capturemode" is mandatory.</td> </tr> <tr> <td>KLS, KLV</td> <td>Parameter "capturemode" is mandatory.</td> </tr> </tbody> </table>	Value	Comment	completed	<ul style="list-style-type: none"> <li>Set with last capture; i.e.: Delivery completed.</li> <li>No further capture is allowed.</li> </ul>	notcompleted	<ul style="list-style-type: none"> <li>Set with partial deliveries (last delivery with "completed")</li> <li>Another capture is expected to complete the transaction.</li> </ul>	Payment type	Comment	PDT	Parameter "capturemode" is mandatory.	KLS, KLV	Parameter "capturemode" is mandatory.
Value	Comment													
completed	<ul style="list-style-type: none"> <li>Set with last capture; i.e.: Delivery completed.</li> <li>No further capture is allowed.</li> </ul>													
notcompleted	<ul style="list-style-type: none"> <li>Set with partial deliveries (last delivery with "completed")</li> <li>Another capture is expected to complete the transaction.</li> </ul>													
Payment type	Comment													
PDT	Parameter "capturemode" is mandatory.													
KLS, KLV	Parameter "capturemode" is mandatory.													
settleaccount	+	<p>Parameter settleaccount is mandatory to indicate whether a refund to the customer should be initiated. (Default: yes)</p> <p>Carry out settlement of outstanding balances. The request is booked and the resulting balance is settled by means of a collection, e.g. a refund. FormatLIST</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Comment</th> </tr> </thead> <tbody> <tr> <td>yes</td> <td>Settlement of outstanding balances is carried out.</td> </tr> <tr> <td>no</td> <td>Do not carry out settlement of outstanding balances, book request only.</td> </tr> <tr> <td>auto</td> <td>The system decides - depending on type of payment and balance - if a settlement of balances can be carried out or not. (default)</td> </tr> </tbody> </table>	Value	Comment	yes	Settlement of outstanding balances is carried out.	no	Do not carry out settlement of outstanding balances, book request only.	auto	The system decides - depending on type of payment and balance - if a settlement of balances can be carried out or not. (default)				
Value	Comment													
yes	Settlement of outstanding balances is carried out.													
no	Do not carry out settlement of outstanding balances, book request only.													
auto	The system decides - depending on type of payment and balance - if a settlement of balances can be carried out or not. (default)													

amount	+	<p>Specifies the total gross amount of a payment transaction.</p> <p>Value is given in smallest currency unit, e.g. Cent of Euro; Pence of Pound sterling; Öre of Swedish krona.</p> <p>The amount must be less than or equal to the amount of the corresponding booking.</p> <p>FormatNUMERIC(1..10) Permitted values max. +/- 19 999 999 99</p>								
currency	+	<p>Carry out settlement of outstanding balances. The request is booked and the resulting balance is settled by means of a collection, e.g. a refund.</p> <p>FormatLIST</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Comment</th> </tr> </thead> <tbody> <tr> <td>yes</td> <td>Settlement of outstanding balances is carried out.</td> </tr> <tr> <td>no</td> <td>Do not carry out settlement of outstanding balances, book request only.</td> </tr> <tr> <td>auto</td> <td>The system decides - depending on type of payment and balance - if a settlement of balances can be carried out or not. (default)</td> </tr> </tbody> </table>	Value	Comment	yes	Settlement of outstanding balances is carried out.	no	Do not carry out settlement of outstanding balances, book request only.	auto	The system decides - depending on type of payment and balance - if a settlement of balances can be carried out or not. (default)
Value	Comment									
yes	Settlement of outstanding balances is carried out.									
no	Do not carry out settlement of outstanding balances, book request only.									
auto	The system decides - depending on type of payment and balance - if a settlement of balances can be carried out or not. (default)									
narrative_text	-	<p>Dynamic text element on account statements</p> <p>FormatCHAR(1..81) (3 lines with 27 characters each) and credit card statements.</p>								
de[n]	-	<p>Description of this item. Will be printed on documents to customer.</p> <p>FormatCHAR(1..255) Array</p> <p><b>Example</b> Array elements [n] starting with [1]; serially numbered; max [400]</p> <pre>de[1]=Product 1 de[2]=Product 2 de[3]=Product 3 ... de[400]=Product 400</pre>								
no[n]	-	<p>Quantity of this item</p> <p>FormatNUMERIC(6) Array</p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p>								
id[n]	-	<p>Product number, SKU, etc. of this item</p> <p>FormatCHAR(1..32) Array</p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p> <p>Permitted Symbols[0-9][A-Z][a-z][ ( ) [ ] { } + - _ # / : ]</p>								

it[n]	-	<p>Parameter it[n] specifies the item type of a shopping cart item.</p> <p>FormatLIST Array</p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p> <table border="1" data-bbox="516 289 1063 615"> <thead> <tr> <th data-bbox="516 289 618 342">it[n]</th> <th data-bbox="618 289 797 342">Comments</th> <th data-bbox="797 289 1063 342"></th> </tr> </thead> <tbody> <tr> <td data-bbox="516 342 618 394">goods</td> <td data-bbox="618 342 797 394">Goods</td> <td data-bbox="797 342 1063 394"></td> </tr> <tr> <td data-bbox="516 394 618 436">shipment</td> <td data-bbox="618 394 797 436">Shipping charges</td> <td data-bbox="797 394 1063 436"></td> </tr> <tr> <td data-bbox="516 436 618 531">handling</td> <td data-bbox="618 436 797 531">Handling fee</td> <td data-bbox="797 436 1063 531"> <ul style="list-style-type: none"> <li>• Not to be used with PDT</li> </ul> </td> </tr> <tr> <td data-bbox="516 531 618 615">voucher</td> <td data-bbox="618 531 797 615">Voucher / discount</td> <td data-bbox="797 531 1063 615"> <ul style="list-style-type: none"> <li>• Not to be used with PDT</li> </ul> </td> </tr> </tbody> </table>	it[n]	Comments		goods	Goods		shipment	Shipping charges		handling	Handling fee	<ul style="list-style-type: none"> <li>• Not to be used with PDT</li> </ul>	voucher	Voucher / discount	<ul style="list-style-type: none"> <li>• Not to be used with PDT</li> </ul>
it[n]	Comments																
goods	Goods																
shipment	Shipping charges																
handling	Handling fee	<ul style="list-style-type: none"> <li>• Not to be used with PDT</li> </ul>															
voucher	Voucher / discount	<ul style="list-style-type: none"> <li>• Not to be used with PDT</li> </ul>															
va[n]	-	<p>VAT rate (% or bp)</p> <p>FormatNUMERIC(4) Array</p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p>															
pr[n]	-	<p>Unit gross price of the item in smallest unit! e.g. cent</p> <p>FormatNUMERIC(10) max. 19 999 999 99Array</p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p>															
add_paydata [shipping_info_shipping_company_n]	-	<p>Name of the shipping company (as specific as possible).</p> <p>FormatCHAR(1..100)SamplesDHL US</p> <p style="text-align: center;">DHL</p>															
add_paydata [shipping_info_shipping_method_n]	-	<p>Shipping method.</p> <p>Permitted ValuesPickUpStore</p> <p style="text-align: center;">Home</p> <p style="text-align: center;">BoxReg</p> <p style="text-align: center;">BoxUnreg</p> <p style="text-align: center;">PickUpPoint</p> <p style="text-align: center;">Own</p>															
add_paydata [shipping_info_tracking_number_n]	-	<p>Tracking number for the shipment.</p> <p>FormatCHAR(1..100)</p>															
add_paydata [shipping_info_tracking_uri_n]	-	<p>URI where the customer can track their shipment.</p> <p>FormatCHAR(1..1024)</p>															
add_paydata [shipping_info_return_shipping_company_n]	-	<p>Name of the shipping company for the return shipment (as specific as possible).</p> <p>FormatCHAR(1..100)SamplesDHL US</p> <p style="text-align: center;">DHL</p>															
add_paydata [shipping_info_return_tracking_uri_n]	-	<p>URL where the customer can track the return shipment.</p> <p>FormatCHAR(1..1024)</p>															

add_paydata [shipping_delay]	-	<p>Delay before the order will be shipped. Use for improving the customer experience regarding payments. This field is currently not returned when reading the order.</p> <p>Please note: to be able to submit values larger than 0, this has to be enabled in your merchant account. Please contact Klarna for further information.</p> <p>FormatINT(1..50)Permitted valuesmin: 0</p>
---------------------------------	---	--

## Response "Capture"

Response "Capture"								
Common Parameter	Required	Comment						
status	+	Permitted ValuesAPPROVED  ERROR						
clearing_instructionnote	o	<ul style="list-style-type: none"> <li>A URL pointing to a PDF of the invoice. (If invoice by mail or by e-mail isn't activated.)</li> <li>The URL is valid for 30 days.</li> <li>This feature has to be requested.</li> </ul>						
Parameter (APPROVED)								
txid	+	The txid specifies the payment process within the PAYONE platform FormatNUMERIC(9..12)						
settleaccount	+	Provides information about whether a settlement of balances has been carried out. FormatLIST <table border="1" data-bbox="490 1283 1068 1476"> <thead> <tr> <th>Value</th> <th>Comment</th> </tr> </thead> <tbody> <tr> <td>yes</td> <td>Settlement of outstanding balances has been carried out.</td> </tr> <tr> <td>no</td> <td>Settlement of outstanding balances has <b>not</b> been carried out.</td> </tr> </tbody> </table>	Value	Comment	yes	Settlement of outstanding balances has been carried out.	no	Settlement of outstanding balances has <b>not</b> been carried out.
Value	Comment							
yes	Settlement of outstanding balances has been carried out.							
no	Settlement of outstanding balances has <b>not</b> been carried out.							
Parameter (ERROR)								
errorcode	+	In case of error the PAYONE Platform returns an error code for your internal usage. FormatNUMERIC(1..6)						
errormessage	-	In case of error the PAYONE Platform returns an error message for your internal usage. FormatCHAR(1..1024)						

customermessage	-	<p>The customermessage is returned to your system in order to be displayed to the customer.</p> <p>(Language selection is based on the end customer's language, parameter "language") FormatCHAR(1..1024)</p>
-----------------	---	---

## Debit

### Request "Debit"

Request "Debit"		
Common Parameter	Required	Short explanation
txid	+	<p>The txid specifies the payment process within the PAYONE platform FormatNUMERIC(9..12)</p>
sequencenumber	+	<p>Sequence number for this transaction within the payment process (1..n), e.g. PreAuthorization: 0, 1. Capture: 1, 2. Capture: 2</p> <p>Required for multi partial capture (starting with the 2<sup>nd</sup> capture) FormatNUMERIC(1..3) Permitted values0..127</p>
amount	+	<p>To transfer an amount from the merchant to the customer, the amount has to be negative.</p> <p>Gross amount of debit (in smallest currency unit! e.g. cent, max. 19 999 999 99)</p> <p>Credit: amount &lt; 0</p> <p>Payment request: amount &gt; 0</p> <p>The amount must be less than or equal to the amount of the outstanding payment request of the corresponding booking. FormatNUMERIC(1..10), max. value +/- 19 999 999 99</p>
currency	+	<p>Specifies currency for this transaction</p> <p>FormatLIST Permitted values ISO 4217 (currencies) 3-letter-codes</p> <p>SamplesEUR USD GBP</p>
narrative_text	-	<p>Dynamic text element on account statements FormatCHAR(1..81) (3 lines with 27 characters each) and credit card statements.</p>

use_custom erdata	-	<p>Use account details from debtor's master data FormatLIST</p> <table border="1" data-bbox="423 222 1065 415"> <thead> <tr> <th>Value</th> <th>Comment</th> </tr> </thead> <tbody> <tr> <td>yes</td> <td>Uses current account details from debtor's master data (default)</td> </tr> <tr> <td>no</td> <td>Uses the last known account details in the payment process</td> </tr> </tbody> </table>	Value	Comment	yes	Uses current account details from debtor's master data (default)	no	Uses the last known account details in the payment process		
Value	Comment									
yes	Uses current account details from debtor's master data (default)									
no	Uses the last known account details in the payment process									
transaction_ param	-	<p>Optional parameter for merchant information (per payment request) FormatCHAR(1..50) Permitted Symbols[0-9][A-Z][a-z][.-_ /]</p>								
settleaccount	+	<p>Parameter settleaccount is mandatory to indicate whether a refund to the customer should be initiated. (Default: yes)</p> <p>Carry out settlement of outstanding balances. The request is booked and the resulting balance is settled by means of a collection, e.g. a refund. FormatLIST</p> <table border="1" data-bbox="423 798 1065 1056"> <thead> <tr> <th>Value</th> <th>Comment</th> </tr> </thead> <tbody> <tr> <td>yes</td> <td>Settlement of outstanding balances is carried out.</td> </tr> <tr> <td>no</td> <td>Do not carry out settlement of outstanding balances, book request only.</td> </tr> <tr> <td>auto</td> <td>The system decides - depending on type of payment and balance - if a settlement of balances can be carried out or not. (default)</td> </tr> </tbody> </table>	Value	Comment	yes	Settlement of outstanding balances is carried out.	no	Do not carry out settlement of outstanding balances, book request only.	auto	The system decides - depending on type of payment and balance - if a settlement of balances can be carried out or not. (default)
Value	Comment									
yes	Settlement of outstanding balances is carried out.									
no	Do not carry out settlement of outstanding balances, book request only.									
auto	The system decides - depending on type of payment and balance - if a settlement of balances can be carried out or not. (default)									
invoiceid	-	<p>Merchant's invoice number FormatCHAR(1..20)</p>								
invoice_deliv erymode	-	<p>Parameter defines how documents like invoice, credit notes and reminders should be sent to the customer. FormatLIST</p> <table border="1" data-bbox="423 1312 678 1501"> <thead> <tr> <th>Value</th> <th>Comments</th> </tr> </thead> <tbody> <tr> <td>M</td> <td>Postal Mail</td> </tr> <tr> <td>P</td> <td>PDF (via email)</td> </tr> <tr> <td>N</td> <td>no delivery</td> </tr> </tbody> </table>	Value	Comments	M	Postal Mail	P	PDF (via email)	N	no delivery
Value	Comments									
M	Postal Mail									
P	PDF (via email)									
N	no delivery									
invoiceappe ndix	-	<p>Dynamic text on the invoice FormatCHAR(1..255)</p>								
invoice_deliv erydate	-	<p>Delivery date (YYYYMMDD) FormatDATE(8), YYYYMMDD</p>								
invoice_deliv eryenddate	-	<p>Delivery end date (YYYYMMDD) FormatDATE(8), YYYYMMDD</p>								



it[n]	-	<p>Parameter it[n] specifies the item type of a shopping cart item.</p> <p>Format <code>LIST Array</code></p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p> <table border="1" data-bbox="423 264 995 590"> <thead> <tr> <th>it[n]</th> <th>Comments</th> <th></th> </tr> </thead> <tbody> <tr> <td>goods</td> <td>Goods</td> <td></td> </tr> <tr> <td>shipment</td> <td>Shipping charges</td> <td></td> </tr> <tr> <td>handling</td> <td>Handling fee</td> <td> <ul style="list-style-type: none"> <li>Not to be used with PDT</li> </ul> </td> </tr> <tr> <td>voucher</td> <td>Voucher / discount</td> <td> <ul style="list-style-type: none"> <li>Not to be used with PDT</li> </ul> </td> </tr> </tbody> </table>	it[n]	Comments		goods	Goods		shipment	Shipping charges		handling	Handling fee	<ul style="list-style-type: none"> <li>Not to be used with PDT</li> </ul>	voucher	Voucher / discount	<ul style="list-style-type: none"> <li>Not to be used with PDT</li> </ul>
it[n]	Comments																
goods	Goods																
shipment	Shipping charges																
handling	Handling fee	<ul style="list-style-type: none"> <li>Not to be used with PDT</li> </ul>															
voucher	Voucher / discount	<ul style="list-style-type: none"> <li>Not to be used with PDT</li> </ul>															
id[n]	-	<p>Product number, SKU, etc. of this item</p> <p>Format <code>CHAR(1..32) Array</code></p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p> <p>Permitted Symbols <code>[0-9][A-Z][a-z][(){}+_-#/:]</code></p>															
pr[n]	-	<p>Unit gross price of the item in smallest unit! e.g. cent</p> <p>Format <code>NUMERIC(10) max. 19 999 999 99 Array</code></p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p>															
no[n]	-	<p>Quantity of this item</p> <p>Format <code>NUMERIC(6) Array</code></p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p>															
de[n]	-	<p>Description of this item. Will be printed on documents to customer.</p> <p>Format <code>CHAR(1..255) Array</code></p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p> <div data-bbox="423 1167 1065 1409" style="border: 1px solid black; padding: 5px;"> <p><b>Example</b></p> <pre> de[1]=Product 1 de[2]=Product 2 de[3]=Product 3 ... de[400]=Product 400 </pre> </div>															
va[n]	-	<p>VAT rate (% or bp)</p> <p>Format <code>NUMERIC(4) Array</code></p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p>															
sd[n]	-	<p>Delivery date (YYYYMMDD)</p> <p>Format <code>DATE(8), YYYYMMDD Array</code></p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p>															
ed[n]	-	<p>Delivery period end date (YYYYMMDD)</p> <p>Format <code>DATE(8), YYYYMMDD Array</code></p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p>															

## Response "Debit"

Response "Debit"								
Common Parameter	Required	Comment						
status	+	Permitted ValuesAPPROVED  ERROR						
Parameter (APPROVED)								
txid	+	The txid specifies the payment process within the PAYONE platform FormatNUMERIC(9..12)						
settleaccount	+	Provides information about whether a settlement of balances has been carried out. FormatLIST <table border="1" data-bbox="490 791 1066 982"> <thead> <tr> <th>Value</th> <th>Comment</th> </tr> </thead> <tbody> <tr> <td>yes</td> <td>Settlement of outstanding balances has been carried out.</td> </tr> <tr> <td>no</td> <td>Settlement of outstanding balances has <b>not</b> been carried out.</td> </tr> </tbody> </table>	Value	Comment	yes	Settlement of outstanding balances has been carried out.	no	Settlement of outstanding balances has <b>not</b> been carried out.
Value	Comment							
yes	Settlement of outstanding balances has been carried out.							
no	Settlement of outstanding balances has <b>not</b> been carried out.							
Parameter (ERROR)								
errorcode	+	In case of error the PAYONE Platform returns an error code for your internal usage. FormatNUMERIC(1..6)						
errormessage	-	In case of error the PAYONE Platform returns an error message for your internal usage. FormatCHAR(1..1024)						
customermessage	-	The customermessage is returned to your system in order to be displayed to the customer.  (Language selection is based on the end customer's language, parameter " <a href="#">language</a> ") FormatCHAR(1..1024)						

## Refund

For refunding / crediting the amount to the customer a Refund request can be triggered.

The refunded amount must not be higher than the captured amount. The refunded amount can be accompanied by a descriptive text and order lines. By using the exact same order line name and reference as in the capture, one will improve the customer's experience in the Klarna App as well the payment instructions sent to the customer.

## Request "Refund"

Request "Refund"								
API parameter	Required	Comments						
txid	+	The txid specifies the payment process within the PAYONE platform FormatNUMERIC(9..12)						
sequencenumber	+	Sequence number for this transaction within the payment process (1..n), e.g. PreAuthorization: 0, 1. Capture: 1, 2. Capture: 2  Required for multi partial capture (starting with the 2 <sup>nd</sup> capture) FormatNUMERIC(1..3) Permitted values0..127						
amount	+	Amount of refund (in smallest currency unit! e.g. cent, max. 19 999 999 99). The amount must be less than or equal to the amount of the corresponding booking. Always provide a negative amount  FormatNUMERIC(1..10), max. value +/- 19 999 999 99						
currency	+	Specifies currency for this transaction  FormatLIST Permitted values ISO 4217 (currencies) 3-letter-codes SamplesEUR USD GBP						
narrative_text	-	Dynamic text element on account statements FormatCHAR(1..81) (3 lines with 27 characters each) and credit card statements.						
use_custom erdata	-	Use account details from debtor's master data FormatLIST <table border="1"> <thead> <tr> <th>Value</th> <th>Comment</th> </tr> </thead> <tbody> <tr> <td>yes</td> <td>Uses current account details from debtor's master data (default)</td> </tr> <tr> <td>no</td> <td>Uses the last known account details in the payment process</td> </tr> </tbody> </table>	Value	Comment	yes	Uses current account details from debtor's master data (default)	no	Uses the last known account details in the payment process
Value	Comment							
yes	Uses current account details from debtor's master data (default)							
no	Uses the last known account details in the payment process							
transaction_param	-	Optional parameter for merchant information (per payment request) FormatCHAR(1..50) Permitted Symbols[0-9][A-Z][a-z][.-_ /]						
invoiceid	-	Merchant's invoice number FormatCHAR(1..20)						

invoice_deliverymode	-	<p>Parameter defines how documents like invoice, credit notes and reminders should be sent to the customer. FormatLIST</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Comments</th> </tr> </thead> <tbody> <tr> <td>M</td> <td>Postal Mail</td> </tr> <tr> <td>P</td> <td>PDF (via email)</td> </tr> <tr> <td>N</td> <td>no delivery</td> </tr> </tbody> </table>	Value	Comments	M	Postal Mail	P	PDF (via email)	N	no delivery
Value	Comments									
M	Postal Mail									
P	PDF (via email)									
N	no delivery									
invoiceappendix	-	<p>Dynamic text on the invoice FormatCHAR(1..255)</p>								
invoice_deliverydate	-	<p>Delivery date (YYYYMMDD) FormatDATE(8), YYYYMMDD</p>								
invoice_deliveryenddate	-	<p>Delivery end date (YYYYMMDD) FormatDATE(8), YYYYMMDD</p>								
de[n]	-	<p>Description of this item. Will be printed on documents to customer. FormatCHAR(1..255) Array</p> <p><b>Example</b> Elements [n] starting with [1]; serially numbered; max [400]</p> <pre>de[1]=Product 1 de[2]=Product 2 de[3]=Product 3 ... de[400]=Product 400</pre>								
no[n]	-	<p>Quantity of this item FormatNUMERIC(6) Array</p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p>								
id[n]	-	<p>Product number, SKU, etc. of this item FormatCHAR(1..32) Array</p> <p>Array elements [n] starting with [1]; serially numbered; max [400] Permitted Symbols[0-9][A-Z][a-z][ ( ) [ ] { } + - _ # / : ]</p>								
pr[n]	-	<p>Unit gross price of the item in smallest unit! e.g. cent FormatNUMERIC(10) max. 19 999 999 99Array</p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p>								

it[n]	-	<p>Parameter it[n] specifies the item type of a shopping cart item.</p> <p>FormatLIST Array</p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p> <table border="1"> <thead> <tr> <th>it[n]</th> <th>Comments</th> <th></th> </tr> </thead> <tbody> <tr> <td>goods</td> <td>Goods</td> <td></td> </tr> <tr> <td>shipment</td> <td>Shipping charges</td> <td></td> </tr> <tr> <td>handling</td> <td>Handling fee</td> <td> <ul style="list-style-type: none"> <li>Not to be used with PDT</li> </ul> </td> </tr> <tr> <td>voucher</td> <td>Voucher / discount</td> <td> <ul style="list-style-type: none"> <li>Not to be used with PDT</li> </ul> </td> </tr> </tbody> </table>	it[n]	Comments		goods	Goods		shipment	Shipping charges		handling	Handling fee	<ul style="list-style-type: none"> <li>Not to be used with PDT</li> </ul>	voucher	Voucher / discount	<ul style="list-style-type: none"> <li>Not to be used with PDT</li> </ul>
it[n]	Comments																
goods	Goods																
shipment	Shipping charges																
handling	Handling fee	<ul style="list-style-type: none"> <li>Not to be used with PDT</li> </ul>															
voucher	Voucher / discount	<ul style="list-style-type: none"> <li>Not to be used with PDT</li> </ul>															
va[n]	-	<p>VAT rate (% or bp)</p> <p>FormatNUMERIC(4) Array</p> <p>Array elements [n] starting with [1]; serially numbered; max [400]</p>															

## Response "Refund"

Response "Refund"								
Common Parameter	Required	Comment						
status	+	Permitted ValuesAPPROVED ERROR						
Parameter (APPROVED)								
txid	+	The txid specifies the payment process within the PAYONE platform FormatNUMERIC(9..12)						
settleaccount	+	Provides information about whether a settlement of balances has been carried out. FormatLIST <table border="1"> <thead> <tr> <th>Value</th> <th>Comment</th> </tr> </thead> <tbody> <tr> <td>yes</td> <td>Settlement of outstanding balances has been carried out.</td> </tr> <tr> <td>no</td> <td>Settlement of outstanding balances has <b>not</b> been carried out.</td> </tr> </tbody> </table>	Value	Comment	yes	Settlement of outstanding balances has been carried out.	no	Settlement of outstanding balances has <b>not</b> been carried out.
Value	Comment							
yes	Settlement of outstanding balances has been carried out.							
no	Settlement of outstanding balances has <b>not</b> been carried out.							
Parameter (ERROR)								
errorcode	+	In case of error the PAYONE Platform returns an error code for your internal usage. FormatNUMERIC(1..6)						

<code>errormessage</code>	-	In case of error the PAYONE Platform returns an error message for your internal usage. FormatCHAR(1..1024)
<code>customermesssage</code>	-	The customermesssage is returned to your system in order to be displayed to the customer.  (Language selection is based on the end customer's language, parameter " <code>language</code> ") FormatCHAR(1..1024)

## Voucher Handling

If you're using vouchers in your checkout, and want to partial refund items, you do have 2 options to do this:

- send a debit-request with the items using the original amount without any discounts and send a additional item as voucher, with an amount that sums up all discounts
  - example cart with auth
    - item 1; red car; 20 €
    - item 2; green truck; 30 €
    - item 3; voucher -10€ (-5€ discount on both items)
  - example cart with debit/refund
    - item 1; red car; 20 €
    - item 2; voucher -5€ (15€ for one item will be returned)
- send a debit-request with the items using an reduced amount, so that the whole discount is broken down on every item.
  - example cart with auth
    - item 1; red car; 20 €
    - item 2; green truck; 30 €
    - item 3; voucher -10€ (-5€ discount on both items)
  - example cart with debit/refund
    - item 1; red car; 15 € (15€ for one item will be returned)