Klarna

Overview

Checkout can be used to accept Klarna payments.

Integration options supported: Hosted payment page (HPP) or IFRAME.

This guide requires familiarity with Accepting payments.

Integrate Klarna via Checkout

Parameters	Type	Required	Description
currency_code	String	Yes	Supported currencies: EUR, DKK, NOK, SEK, USD
amount	Integer	Yes	Transaction amount
configurations.klarna	Object	Yes	Object carrying the parameters required for making a Klarna payment
customer	String	Yes	ID of a Customer created via the Customer API. A customer object can be created and attached to a Checkout. Supported Countries ('US', 'NO', 'SE', 'FI', 'DK')
capture_now	Boolean	No	Used for separate authorization and capture
line_items	Array of objects	Yes	Each line item represents a product in the checkout

Authorization and capture

Klarna transactions through the Checkout can be used to do a sale (capture_now = true) or to authorize without capturing immediately (capture_now = false). An authorized payment reserves the money and allows you to capture the funds at a later stage.

Handling responses

Whenever a card payment is processed via the Checkout, the responses events would contain additional fields specific to card payments in the **details** object.

Sending a checkout request using Klarna payment method:

```
{
    "amount": 12054,
    "currency_code": "EUR",
    "entity_id": "030288dd-97fd-449e-9753-1a8195d92c56",
    "customer": "53fc359e-7cb0-4bd5-94d6-8f7f2ef48511",
    "configurations": {
        "klarna" : {
            "capture_now": false
        }
    },
    "line_items": [
        {
            "name": "Demo product",
            "quantity": 1,
            "tax_rate": 0,
            "total_tax_amount": 0,
            "total_amount": 12054,
            "unit_price": 12054
        }
    ],
    "merchant_reference": "ORDER-9671",
    "return_url": "https://enzxiezwyvbcusi.m.pipedream.net",
    "interaction_type": "HPP"
}
```

Example of successful checkout via the Checkout:

```
[
    {
        "type": "TRANSACTION_INITIATED",
        "id": "a4eb96fd-f448-420c-b2f8-e8dd758513c4",
        "timestamp": "2021-09-23T12:08:38.970Z",
        "details": {
            "id": "551deb6b-1bd1-46e6-bf36-a18555212b1a",
            "instore_reference": "9123285100004542",
            "client_token": "eyJh******ZaoA",
            "payment_method_categories": [
                {
                    "identifier": "pay_over_time",
                    "name": "Slice it.",
                    "asset_urls": {
                        "standard":
"https://x.klarnacdn.net/payment-method/assets/badges/generic/klarna.svg",
                        "descriptive":
"https://x.klarnacdn.net/payment-method/assets/badges/generic/klarna.svg"
                    }
                },
                {
                    "identifier": "pay_later",
                    "name": "Pay later.",
                    "asset_urls": {
                        "standard":
"https://x.klarnacdn.net/payment-method/assets/badges/generic/klarna.svg",
                        "descriptive":
```

```json
                "https://x.klarnacdn.net/payment-method/assets/badges/generic/klarna.svg"
                    }
                },
                {
                    "identifier": "pay_now",
                    "name": "Pay now.",
                    "asset_urls": {
                        "standard":
"https://x.klarnacdn.net/payment-method/assets/badges/generic/klarna.svg",
                        "descriptive":
"https://x.klarnacdn.net/payment-method/assets/badges/generic/klarna.svg"
                    }
                }
            ],
            "amount": 11703,
            "blocked": false,
            "customer": "53fc359e-7cb0-4bd5-94d6-8f7f2ef48511",
            "details": {
                "auto_capture": false
            },
            "merchant_reference": "ORDER-5632",
            "processor": "KLARNA",
            "payment_product": "KLARNA",
            "payment_product_type": "Klarna",
            "status": "INITIATED",
            "created_by": "cfc403f7-95a3-4c84-b9fd-2e18c37e488d",
            "shopper_interaction": "ECOMMERCE",
            "geo_location": [
                26.9261,
                75.7928
            ],
            "city": "Jaipur",
            "country_code": "FI"
        }
    },
    {
        "type": "TRANSACTION_SUCCESS",
        "id": "7ce36c1e-bc31-4958-bb30-fd3f52a6291f",
        "timestamp": "2021-09-23T12:13:00.514Z",
        "details": {
            "id": "551deb6b-1bd1-46e6-bf36-a18555212b1a",
            "instoreReference": "9123285100004542",
            "authorizedPaymentMethod": {
                "paymentType": "INVOICE"
            },
            "amount": 11703,
            "createdAt": "2021-09-23T12:08:38+01:00",
            "details": {
                "autoCapture": false,
                "mid": "K043993"
            },
            "merchantReference": "ORDER-5632",
            "processor": "KLARNA",
            "paymentProduct": "KLARNA",
            "paymentProductType": "Klarna",
            "shippingInformation": {},
```

```
            "status": "AUTHORIZED",
            "shopperInteraction": "ECOMMERCE",
            "countryCode": "FI"
        }
    }
]
```

Example of failed Klarna payment via the Checkout:

```
[
    {
        "type": "TRANSACTION_FAILED",
        "id": "63fb2e62-4d16-4f85-9886-7f66f7f09ab1",
        "timestamp": "2021-09-23T12:46:48.404Z",
        "details": {
            "code": 404,
            "details": {
                "error":
"404 : [{\n  \"error_code\" : \"NOT_FOUND\",\n  \"error_messages\" : [ \"Invalid
,
                "serviceCode": "KLARNA"
            },
            "timestamp": 1632401208387,
            "message":
"The requested resource, or one of its sub-resources, can't be found. If the subm

        }
    }
]
```

To ensure that the redirection request was not tampered with, always check that the transaction_id received as query parameter in the redirection matches the transaction_id property of the retrieved Checkout. If those are not matching, this is indication of either an incorrect integration, that the redirection to your return_url did not originate from Verifone, or transaction_id was tampered with.

You can now store the transaction_id value together with the order 1234 in your system to link the two together.

Scenarios

The table below describes the different outcomes of a Checkout. A full list of error codes are available.

Description	Result	Merchan action
Failed transaction*	Redirect: checkout_id={checkout_id} & transaction_id={transaction_id} & errorCode=123	Unsuccessful payment (technical reason), do not display order confirmation
Successful transaction	Redirect: checkout_id={checkout_id} & transaction_id={transaction_id}	Display order confirmation

Description	Result	Merchan action
Customer visits the URL of an already completed Checkout	Redirect: checkout_id={checkout_id} & errorCode=168	Display corresponding message to the customer. Checkout is completed whenever there was a single successful payment processed through it.
Customer visits the URL of an expired Checkout	Redirect: checkout_id={checkout_id} & errorCode=169	Display corresponding message to the customer. Checkout is expired whenever the expiry_time is reached
Customer visits the URL of a Checkout which has reached the maximum of failed payment attempts	Redirect: checkout_id={checkout_id} & errorCode=165	Display corresponding message to the customer. Payments through a single Checkout can be attempted up to three times unsuccessfully.
Form validation errors / Other service failures on the Checkout page	Displays error alert to Customer on the page	Customer is prompted to correct their form input and retry the payment or try using alternate card or the payment method

* Failed transaction - Depending on which step in the payment process failed, the transaction_id might not always be present as the query parameter.