Vipps

Overview

Checkout can be used to accept Vipps payments.

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

This guide requires familiarity with Accepting payments.

Integrate Vipps via Checkout

Parameters	Type	Required	Description
currency_code	String	Yes	Supported currencies: NOK
amount	Integer	Yes	Transaction amount
customer	String	Yes	ID of a Customer created via the Customer API. A customer object can be created and attached to a Checkout.
configurations.vipps	Object	Yes	Object carrying the parameters required for making a Vipps payment
payment_contract_id	String	Yes	This ID can be found in the portal or given to you by a Verifone employee. It is used to retrieve a merchant's Vipps credentials necessary for payment.
dynamic_descriptor	String	No	Short text to be displayed on the bank statement of the cardholder. Support varies per Payment Contract.
capture_now	Boolean	No	Used for separate authorization and capture
is_app	Boolean	No	Flag to identify the transaction originated from APP or Browser
app_phone_number	String	No	Indicates the phone number, without country code, registered with Vipps Mobile APP. Supported value: 8 digits without spaces.

Parameters	Type	Required	Description
`card`	Object	No	Details regarding SCA compliance **Required**, 3DS **Required** and authorization type **Required**
`authorization_type`	String	No	Card Authorization Type (PRE_AUTH, FINAL_AUTH). When capture_now is set to true, pre-authorization transactions cannot be done.

Authorization and capture

You can use Vipps transactions through the Checkout 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.

Sending a checkout request using Vipps payment method:

```
{
"currency_code": "NOK",
"entity_id": "466c5c59-5177-45c2-b9a7-41ac422ef1fa",
"customer": "deebacec-cd75-4ce5-9641-bfc40de5710d",
"configurations": {
"vipps": {
"dynamic_descriptor": "Test Product",
"capture_now": true,
"payment_contract_id": "98a003be-03f2-43c4-a89c-36021fa63635",
"app_phone_number": "40950037",
"card": {
"sca_compliance_level": "WALLET",
"authorization_type": "FINAL_AUTH",
"threed_secure": {
"enabled": true,
"threeds_contract_id": "76c87838-610e-40cf-8434-c91debb7cfd9",
"total_items": "01",
"transaction_mode": "S"
}
}
}
},
"amount": 4300,
"merchant_reference": "Test",
"return_url": "http://enbyhy7yz2lg98r.m.pipedream.net",
"interaction_type": "HPP"
}
```

Handling responses

Whenever a card payment is processed via the Checkout, the responses events include additional parameters specific to card payments in the **Details** object.

Example of a successful Vipps payment via the Checkout:

```
[
    {
        "type": "TRANSACTION_INITIATED",
"id": "8de44848-c9eb-48c1-bafe-ba97bf3fb77f",
        "timestamp": "2021-09-15T12:03:55.368Z",
        "details": {
            "amount": 400,
            "created_at": "2021-09-15T12:03:55.071859Z",
            "customer": "deebacec-cd75-4ce5-9641-bfc40de5710d",
            "details": {},
            "merchant_reference": "f811eb59-9fb6-4ac0-a0ee-df36e8e0fe4f",
            "status": "INITIATED",
            "shopper_interaction": "ECOMMERCE",
            "geo_location": [
                20.0063,
                77.006
            ],
            "country_code": "FI",
            "id": "03732287-5fca-4e63-baa7-9f059af58a62",
            "redirect_url": "https://api.vipps.no/dwo-api-
application/v1/deeplink/vippsgateway?v=2&token=eyJraWQiOiJqd3RrZXkiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJmMDE0MmIxYy02
YjIwLTQ1M2MtYTlmMS1lMWUwZGFiNjkzOTciLCJubyI6eyJoZWFkZXIiOiJCZXRhbCIsImNvbnRlbnQiOiI0IGtyIHRpbCBURVNUVklQUFNFQ09N
MSJ9LCJtb2IiOiI0MDk1MDAzNyIsImN0eCI6ImM0YjZhNWM5LWI4ODktNGRhMC05MWNmLTAxYjQ4MDU4MDI4ZSIsIm1lcmNoYW50U2VyaWFsTnVt
YmVyIjoiMjI2NTYzIiwiaXNzIjoiaHR0cHM6XC9cL1ZJUFBTLU1ULUNPTi1BR0VOVC1pbGIudGVjaC0wMi5uZXRcL210MVwvZGVlcGxpbmstb3Bl
bmlkLXByb3ZpZGVyLWFwcGxpY2F0aW9uXC8iLCJlbiI6eyJoZWFkZXIiOiJQYXkiLCJjb250ZW50IjoiNCBrciB0byBURVNUVklQUFNFQ09NMSJ9
LCJ0eXBlIjoiUFNQX1BBWU1FTlQiLCIxZmEiOiJ0cnVlIiwidXVpZCI6ImQzMGUxZTViLTg3MTgtNDFiMC04Y2ZjLTViNDM3ZTBjMzZiYyIsInVy
bCI6Imh0dHBzOlwvXC9hcGl0ZXN0LnZpcHBzLm5vXC9sYW5kaW5ncGFnZVwvZHdvXC92MVwvbGFuZGluZ3BhZ2UiLCJyZWZlcmVuY2VJZCI6ImY4
MTFlYjU5LTlmYjYtNGFjMC1hMGVlLWRmMzZlOGUwZmU0ZiIsImF1ZCI6ImYwMTQyYjFjLTZiMjAtNDUzYy1hOWYxLWUxZTBkYWI2OTM5NyIsImF6
cCI6ImYwMTQyYjFjLTZiMjAtNDUzYy1hOWYxLWUxZTBkYWI2OTM5NyIsImFwcFR5cGUiOiJMQU5ESU5HUEFHRSIsIm5vRWRpdCI6ZmFsc2UsImV4
cCI6MTYzMTcwNzczNSwidG9rZW5UeXBlIjoiREVFUExJTksiLCJpYXQiOjE2MzE3MDc0MzUsImZhbGxiYWNrIjoiaHR0cHM6XC9cL2Rldi5jaGVj
a291dC52Zmltcy5jb21cL3YyXC9jaGVja291dFwvYWIwYTEzNGEtYjI5Zi00YzhjLThjYWQtOTIzMjkwOTkzNGFmXC9yZXR1cm4iLCJqdGkiOiIz
MGZjOTczNy1jY2NiLTRhNDgtYWJiZS0zMjdkM2MwOWU2OTMifQ.Wt48XLuOu0t_2J8xKxaesU9fgjtsWGOEy7mPeZ0Sy-
A_sjTpV41rUZOUwtJ7zvPc5LHAPr0FbhUAFdN4OBZP54K6X_SE0_Z4xx6GMJOGSj17IRqD2FU01APd7P65kWf5yRj9r3yj8gXpmQmfOF-
ASCK9On9-P1_Ki35tcYJNERcjqiGUhEd_1D1OczPWjQv-9Kg2D6AtkMz3aWXbftmakiKHLsBx42groPCLYqCAdxl5xvpVyBc-
X2q19IryJ3mIwpHasAZGUEcek2jkoS0kGCVbkBu_gj33yfL6Igr8dxrbj4klLOXk7sMO_bUEf8U-51RiGs8y6XXkScSDyWR3fQ",
            "processor": "NETS",
            "payment_product": "CARD",
            "payment_product_type": "Unknown"
        }
    },
    {
        "type": "TRANSACTION_SUCCESS",
        "id": "1aaebd63-2daf-4753-bc32-bcdb8c35b743",
        "timestamp": "2021-03-31T11:44:07.717Z",
        "details": {
            "id": "fd31890c-8c45-4846-a934-a7faa584e823",
            "captureId": "93459848W31137123",
            "createdAt": "2021-03-31T11:44:06Z",
            "isFinalCapture": true,
            "payer": {
                "payerId": "92FT3NN28KELQ",
                "shippingAddress": {
                    "country": "IN",
                    "postalCode": "100047",
                    "countrySubdivision": "Maharashtra",
                    "city": "Mumbai",
                    "addressLine1": "addressLine1",
                    "addressLine2": "addressLine2",
                }
            },
            "status": "AUTHORISED"
        }
    }
]
```

Example of a failed Vipps payment via the Checkout:


```
[
    {
        "type": "TRANSACTION_FAILED",
        "id": "f2041250-4fc2-4b3a-bc94-651ba099541a",
        "timestamp": "2020-07-08T12:42:37.974Z",
        "details": {
              "id": "6d85d047-1e78-4f13-afe2-cc041bce524f",
              "authorizationId": "4PB16132MH590854B",
              "createdAt": "2021-01-06T04:40:30Z",
              "expiresAt": "2021-02-04T04:40:30Z",
              "status": "FAILED",
              "payer": {
                  "payerId": "92FT3NN28KELQ",
                  "shippingAddress": {
                      "country": "IN",
                      "postalCode": "100047",
                      "countrySubdivision": "Maharashtra",
                      "city": "Mumbai",
                      "addressLine1": "addressLine1",
                      "addressLine2": "addressLine2",
                  }
        }
      }
    }
  ]
```


- 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 is available.

Description	Result	Merchant 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	Merchant 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 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 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