

## Webhooks

### Introduction

Webhooks are used to send a notification about an update to a transaction, refund or chargeback. In payments it is important for you (merchant) to stay up-to-date with the status of your transactions and webhooks enable them to automate through applications.

### How to implement

You have the option to apply a webhook on a transaction or on an account, depending on the webhook tool used.

When you apply on the account level, you will receive an update for every transaction/refund/chargeback attempt using that specific account. Note that you can only set up the webhook notification for Refunds and Chargebacks on your Account level and not on the individual element.

The "webhook\_transaction\_update" field is used to store the webhook URL whether you want the webhook update on transaction or account level. Below is an example transaction and account request with a webhook url:

### Retries

The webhooks will retry after 1, 8, 27, 64 and 125 minutes if the initial webhook couldn't be delivered.

### Transaction Webhook

```
{
  "account": "05b6f7431e9e4867ac203419",
  "amount": 1,
  "card": "f7d69755d6e3533d8526428d",
  "customer_ip": "127.0.0.1",
  "customer": "00022a399ddb489elda8b515",
  "dynamic_descriptor": "Testing0612345678",
  "merchant_reference": "DemoRef001",
  "payment_product": "card",
  "capture_now": false,
  "user_agent": "Opera/12.02 (Android 4.1; Linux; Opera Mobi/ADR-1111101157; U; en-US) Presto/2.9.201
Version/12.02",
  "webhook_transaction_update": "https://webhook.site/03a24fde-1369-4b86-82fe-e48afc4b2e1f"
}
```

### Account Webhook

```
{
  "currency_code": "EUR",
  "name": "Helsinki Account 1",
  "organisation": "be4eaceb0a4eab1c4fc3d1bb",
  "disabled": false,
  "processors": ["eb69c095225f701d124c3f09"],
  "webhook_transaction_update": "https://webhook.site/03a24fde-1369-4b86-82fe-e48afc4b2e1f"
  "webhook_refund_update": "https://webhook.site/03a24fde-1369-4b86-82fe-e48afc4b2e1f"
  "webhook_chargeback_update": "https://webhook.site/03a24fde-1369-4b86-82fe-e48afc4b2e1f"
}
```

Once you have chosen how you will receive the notification, you will need to handle the webhook response. The response will include the ID and the type (transaction, refund or chargeback). You will need to query the ID to fetch the updated status. Below is an example webhook

response for a transaction:

```
{
  "_id": "54884a22e1e6573d1d1ee001",
  "type": "transaction"
}
```

## When to use

In general, you can use webhooks for services that are not directly responsible for making an API request, but you still need to know the response from that request. Chargeback and Refunds are two of the common uses for webhooks. [Here](#) is the link to the documentation for creating webhooks in our environment.

For Alternative Payment Methods like iDEAL, webhooks are used in order to receive updates when there's a status change for the transaction. An example for iDEAL is a webhook used when the customer completes the checkout from the iDEAL form.

For card payments, webhooks can be used to notify when the transaction is settled, since this normally happens the next day. We also recommend using webhooks for card payments utilizing our Checkout page for situations such as when the customer aborts the redirection flow. In this case, if you are set up with webhooks, you will be updated as soon as a transaction is sent.