



Overview

YowPay account gives you the opportunity to sell better your products and manage easily your finance via API or a merchant portal.

We do automatic reconciliation.

We support instant SEPA payments

We provide detailed reporting of funds movement.

Configure your business

You must configure your business in our merchant portal


1. Login with your account - <https://merchantv2.yowpay.com/login>
2. Go to Business menu and add a new Business or edit already existing one (**name and url** are required). Your credentials will be automatically generated (**app token + secret key**). !!! Remember - **never expose the secret key** !!! The credentials can be used for making API calls and receiving webhooks.
3. We strongly recommend to configure also the additional URLs that will activate the webhooks and will improve the experience of your customers

Return URL : is the url on which Yowpay will redirect the client after she/he confirms the submission of the payment (payment is not necessary credited yet).

Cancel URL : is the url on which Yowpay will redirect the client in case he refuses to make the payment, this redirection is usually used for cascading purposes.

Webhook URL : url on merchant's web site to receive confirmation of accepted transaction. Usually this url is used to credit the end users of the services ordered or validate an order.

CREDENTIALS

App Token
tXSYk+qPvCICzHZDWT+B... 

App Secret Key
1QRC3iGgrPqyfjmfCBCQxSLzZJj+d8OISVMfBb... 

[Generate new Secret Key and Token](#)

Back to website URL

Cancel URL


Webhook URL

Start selling your product

In order to start selling, you must create `payment requests` via an API. It contains the amount, currency, orderId of your product. It may contain additional features - see full API documentation for more details. We reply with URL of payment page that offers different ways of payments.

Choose a method to make your payment


Scan the QR Code
from your mobile banking app if compatible





How to scan? Download QR code


Select your bank
and pay by instant transfer


France ▾ Name of the bank



Crédit Mutuel









BNP Paribas


Société Générale


La Banque Postale


CIC Banques


Boursobank

-  Allianz Banque
-  Arkéa Banque Privée
-  Arkéa Banking Services
-  Arkéa Banque Entreprises et Institutionnels
-  AXA Banque
-  Banque BCP
-  Banque Chalus

Manual Instant Transfer
using the bank details below

Beneficiary

IBAN

Unique Payment Reference
YPRH NNTC
Mandatory input to process your payment

Amount
€33.77

Additional Information
If requested by your bank

BIC / SWIFT

Address

Zip Code

City

Country

[Back to website](#)

Once the order is paid, we send a webhook (to URL you already configured) with transaction details.

Pay attention that there is a difference between the `payment request` (the request created by you) and the actual transaction (when funds are received in the bank account).

Yowpay API manual and webhooks

API calls

All API calls require like a **minimum** the following **headers** and **body** in order to do correct authentication.

- **headers**

Content-type - application/json

X-App-Access-Ts - the timestamp when the call is initiated (UTC must be used)

X-App-Token - the **app token** (part of E-Commerce credentials explained above)

X-App-Access-Sig - the signature of the body (hash data). In order to receive you must use the **secret key** from credentials to hash the body content of the call

```
hash_hmac("sha256", "BodyWithParameters", "YOUR_SECRET_KEY");
```

- the **body** must contain the following parameter in json format

timestamp - the same timestamp that is set in the header (required - int)

– **Check connection and credentials**

config/checkConnection

The purpose of this call is to verify you set the credentials and calculate hash correctly. No action is done. In order to verify the call - no error should be displayed.

It is a POST request - endpoint is <https://api.yowpay.com/config/checkConnection>

- **headers**

X-App-Access-Ts - the timestamp when the call is initiated (UTC must be used)

X-App-Token - the **app token** (part of E-Commerce credentials explained above)

X-App-Access-Sig - the signature of the body (hash data). In order to receive you must use the **secret key** from credentials to hash the body content of the call

- **body**

timestamp - the same timestamp that is set in the header (required - int)

Sample request:

```
{"timestamp":1761659907}
```

Sample response in **json** format:

In case of **error** it returns **http code 400** for Bad Request:

```
{  
  "error": {  
    "code": 132057,  
    "msg": "Invalid data"  
  }  
}
```

In case of **success** it returns **http code 200**

```
{  
  "content": {  
    "success": 1  
  }  
}
```

}

– **Create payment request**

payment/request

It is a POST request - endpoint is <https://api.yowpay.com/payment/request>

- headers

X-App-Access-Ts - the timestamp when the call is initiated (UTC must be used)

X-App-Token - the **app token** (part of E-Commerce credentials explained above)

X-App-Access-Sig - the signature of the body (hash data). In order to receive you must use the **secret key** from credentials to hash the body content of the call

- body

- **amount** - (required - string - only numbers and a dot for decimal separator are allowed - example - 1234.56)
- **currency** - EUR (required - string)
- **timestamp** - the same timestamp that is set in the header (required - int)
- **orderId** - end user reference (optional - string)
- **language** - end user language selection (optional - string - 2 letters)
- **returnUrl** - dynamically set URL; it overwrites the return URL configured for the business (optional - string)
- **cancelUrl** - dynamically set URL; it overwrites the cancel URL configured for the business (optional - string)
- **clientType** - end user type (optional - int), 1 - individual, 2 - company
- **clientFirstName** - end user first name only if clientType = 1 (optional - string)
- **clientLastName** - end user last name only if clientType = 1 (optional - string)
- **clientCompanyName** - end user company name only if clientType = 2 (optional - string)
- **paymentMethodPis** - forcing hiding or displaying a payment method to the end user (optional - int), 1 - display PIS option, 2 - hide PIS option
- **paymentMethodQrCode** - forcing hiding or displaying a payment method to the end user (optional - int), 1 - display QR Code option, 2 - hide QR Code option
- **paymentMethodManual** - forcing hiding or displaying a payment method to the end user (optional - int), 1 - display Manual option, 2 - hide Manual option

The "clientType" and the following variables are optional. Merchants should provide them if available, as this avoids requesting the same details during the PIS flow (which some banks require as mandatory). Supplying them upfront helps ensure a smoother payment process.

Response fields:

- **paymentPageUrl** - (url to a payment page)
- **paymentRequestId** - (payment request identifier)

Sample request:

```
{ "amount": "12.34", "currency": "EUR", "orderId": "test payment method", "clientType": 1, "clientCompanyName": "My Business Ltd", "paymentMethodPis": 1, "paymentMethodManual": 2, "timestamp": 1675772490 }
```

Sample response:

Version 1.25 (20 Mar, 2026)

In case of **error** it returns: (**http code 400**)

```
{
  "error": {
    "code": 132057,
    "msg": "Invalid data"
  }
}
```

In case of **success** it returns the url to the payment page (**http code 200**)

```
{
  "content": {
    "success": 1,
    "paymentPageUrl": "https://secure.yowpay.com/request/payment/1234567/YPABCD",
    "paymentRequestId": "1234567"
  }
}
```

– **Check payment request**

paymentRequest/check

It is a POST request - endpoint is <https://api.yowpay.com/paymentRequest/check>

- **headers**

X-App-Access-Ts - the timestamp when the call is initiated (UTC must be used)

X-App-Token - the **app token** (part of E-Commerce credentials explained above)

X-App-Access-Sig - the signature of the body (hash data). In order to receive you must use the **secret key** from credentials to hash the body content of the call

- **body**

- **paymentRequestId** - (required)
- **timestamp** - the same timestamp that is set in the header (required - int)

paymentRequestId can be found when payment request is created or when a webhook is received.

Response fields:

- **id** - (paymentRequestId)
- **amount** - (!!! this is the amount of the payment request - it can be different from an actual payment !!! example - 12.34)
- **currency** - (!!! this is the currency of the payment request - it can be different from an actual payment !!! 3 letters)
- **statusCode** (it can be **0** for still **pending**, **1** for payment **credited** with expected amount, **2** for payment **credited** but **amount and currency don't match** with the original payment request)

- **amountPaid** - (!!! this is the amount of the actual funds received - it can be different from the amount of the original payment request!!! example - 12.34) - it can be empty (0.00) if still pending
- **currencyPaid** -(!!! this is the currency of the actual funds received - it can be different from the currency of the original payment request!!! 3 letters)- it can be empty if still pending
- **orderId**
- **reference** (YowPay reference)
- **language** (2 letters)
- **creationDate** (payment request creation time - format ISO-8601 - e.g. 2025-02-15T10:40:24Z)
- **validateDate** (transaction validation time - format ISO-8601 - e.g. 2025-02-15T10:40:24Z) - it can be empty if still pending
- **senderIban** - it can be empty if still pending
- **senderAccountHolder** - (account holder name of the sender) it can be empty if still pending

Sample request:

```
{"paymentRequestId":123456789,"timestamp":987654321}
```

Sample response:

In case of **error** it returns: (**http code 400**)

```
{
  "error": {
    "code": 132057,
    "msg": "Invalid data"
  }
}
```

In case of **success** it returns the url to the payment page (**http code 200**)

```
{
  "content": {
    "success": 1,
    "paymentRequestData": {
      "id": 123456789,
      "amount": "45.67",
      "currency": "EUR",
      "amountPaid": "45.67",
      "currencyPaid": "EUR",
      "statusCode": 1,
      "reference": "YPRDQCPC",
      "language": "fr",
      "orderId": "BILLID_123647789",
      "creationDate": "2025-12-23T20:32:07+01:00",

```

```
"validateDate": "2025-12-23T20:32:47+01:00",  
"senderIban": "NL24ABNA8502137913",  
"senderAccountHolder": "John Smith"  
}  
}  
}
```

– Get transaction data

transaction/get

It is a POST request - endpoint is <https://api.yowpay.com/transaction/get>

- headers

X-App-Access-Ts - the timestamp when the call is initiated (UTC must be used)

X-App-Token - the **app token** (part of E-Commerce credentials explained above)

X-App-Access-Sig - the signature of the body (hash data). In order to receive you must use the **secret key** from credentials to hash the body content of the call

- body

- **transactionId** - (required)
- **timestamp** - the same timestamp that is set in the header (required - int)

transactionId can be found when a webhook is received.

Response fields:

- **id** - (transactionId)
- **paymentRequestId**
- **amount** - (the amount of the funds received, example - 12.34)
- **currency** - (the currency of the funds received, 3 letters)
- **statusCode** (**1** for payment **credited**, **9** for payment **declined**)
- **typeCode** (**1** for **credited**, **3** for **refund**)
- **creationDate** (transaction creation time - format ISO-8601 - e.g. 2025-02-15T10:40:24Z)
- **actionDate** (transaction update time - format ISO-8601 - e.g. 2025-02-15T10:40:24Z)
- **senderIban**
- **senderAccountHolder** - (account holder name of the sender)
- **beneficiaryIban**
- **beneficiaryAccountHolder** - (account holder name of the recipient)
- **paymentReference** (test that should appear on bank statement)
- **bankReference**
- **isCreditTransaction** (**0** for **debit**, **1** for **credit**)
- **feeData**
 - **amount** - (the amount of the fee charged, example - 0.50)
 - **currency**
 - **creationDate**

Sample request:

```
{"transactionId":444555666,"timestamp":987654321}
```

Version 1.25 (20 Mar, 2026)

Sample response:

In case of **error** it returns: (**http code 400**)

```
{
  "error": {
    "code": 132057,
    "msg": "Invalid data"
  }
}
```

In case of **success** it returns the url to the payment page (**http code 200**)

```
{
  "content": {
    "success": 1,
    "transactionData": {
      "id": 444555666,
      "paymentRequestId": 99887766,
      "amount": "45.67",
      "currency": "EUR",
      "statusCode": 1,
      "typeCode": 1,
      "creationDate": "2025-12-23T20:32:07+01:00",
      "actionDate": "2025-12-23T20:32:47+01:00",
      "senderIban": "NL24ABNA8502137913",
      "senderAccountHolder": "John Smith",
      "beneficiaryIban": "NL26RABO6798610471",
      "beneficiaryAccountHolder": "My company Ltd",
      "paymentReference": "Gift for anniversary",
      "bankReference": "9dcc59ab-fa74-41df-ae90-f053112e7088",
      "isCredittransaction": 1,
      "feeData": [
        {
          "amount": "0.50",
          "currency": "EUR",
          "creationDate": "2025-12-23T20:32:07+01:00"
        }
      ]
    }
  }
}
```

```
}  
}  
}
```

– Get transaction list

transaction/list

It is a POST request - endpoint is <https://api.yowpay.com/transaction/list>

- headers

X-App-Access-Ts - the timestamp when the call is initiated (UTC must be used)

X-App-Token - the **app token** (part of E-Commerce credentials explained above)

X-App-Access-Sig - the signature of the body (hash data). In order to receive you must use the **secret key** from credentials to hash the body content of the call

- body

- **timestamp** - the same timestamp that is set in the header (required - int)
- **dateFrom** - Format 'YYYY-MM-DD'; date is included in the search (required)
- **dateTo** - Format 'YYYY-MM-DD'; date is included in the search (required)
- **nextData** - In case of more than 10 000 results we need to execute a new call with this data to fetch second page, third page etc (optional)

Response fields:

- **id** - (transactionId)
- **paymentRequestId**
- **amount** - (the amount of the funds received, example - 12.34)
- **currency** - (the currency of the funds received, 3 letters)
- **statusCode** (**1** for payment **credited**, **9** for payment **declined**)
- **typeCode** (**1** for **credited**, **3** for **refund**)
- **creationDate** (transaction creation time - format ISO-8601 - e.g. 2025-02-15T10:40:24Z)
- **actionDate** (transaction update time - format ISO-8601 - e.g. 2025-02-15T10:40:24Z)
- **senderIban**
- **senderAccountHolder** - (account holder name of the sender)
- **beneficiaryIban**
- **beneficiaryAccountHolder** - (account holder name of the recipient)
- **paymentReference** (test that should appear on bank statement)
- **bankReference**
- **isCreditTransaction** (**0** for **debit**, **1** for **credit**)
- **feeData**
 - **amount** - (the amount of the fee charged, example - 0.50)
 - **currency**
 - **creationDate**

We return also "nextData" fields. If it is not empty, we may use it to fetch second page of transactions

Sample request:

```
{"dateFrom":"2025-12-23","dateTo":"2025-12-23","timestamp":987654321}
```

Sample response:

In case of **error** it returns: (**http code 400**)

```
{
  "error": {
    "code": 132057,
    "msg": "Invalid data"
  }
}
```

In case of **success** it returns the url to the payment page (**http code 200**)

```
{
  "content": {
    "success": 1,
    "transactionData": [
      {
        "id": 444555666,
        "paymentRequestId": 99887766,
        "amount": "45.67",
        "currency": "EUR",
        "statusCode": 1,
        "typeCode": 1,
        "creationDate": "2025-12-23T20:32:07+01:00",
        "actionDate": "2025-12-23T20:32:47+01:00",
        "senderIban": "NL24ABNA8502137913",
        "senderAccountHolder": "John Smith",
        "beneficiaryIban": "NL26RABO6798610471",
        "beneficiaryAccountHolder": "My company Ltd",
        "paymentReference": "Gift for anniversary",
        "bankReference": "9dcc59ab-fa74-41df-ae90-f053112e7088",
        "isCreditttransaction": 1,
        "feeData": [
          {
            "amount": "0.50",
            "currency": "EUR",

```

```
        "creationDate": "2025-12-23T20:32:07+01:00"
    }
}
},
"nextData": "123456"
}
}
```

– Get bank account balance data

bankAccount/getBalance

It is a POST request - endpoint is <https://api.yowpay.com/bankAccount/getBalance>

- headers

X-App-Access-Ts - the timestamp when the call is initiated (UTC must be used)
X-App-Token - the **app token** (part of E-Commerce credentials explained above)
X-App-Access-Sig - the signature of the body (hash data). In order to receive you must use the **secret key** from credentials to hash the body content of the call

- body

- **timestamp** - the same timestamp that is set in the header (required - int)

Response fields:

- **iban**
- **currency** - (the currency of the bank account, 3 letters)
- **totalAmount** (the total balance of the bank account, example - 10291.11)
- **reservedAmount** (the blocked amount - for example - for pending payout)
- **availableAmount** (the difference between the total amount and the reserved amount)

Sample request:

```
{"timestamp":987654321}
```

Sample response:

In case of **error** it returns: (**http code 400**)

```
{
  "error": {
    "code": 132057,
    "msg": "Invalid data"
  }
}
```

```
}
```

In case of **success** it returns the url to the payment page (**http code 200**)

```
{
```

```
  "content": {
```

```
    "success": 1,
```

```
    "balanceData": {
```

```
      "iban": "NL26RABO6798610471",
```

```
      "currency": "EUR",
```

```
      "totalAmount": "10291.22",
```

```
      "reservedAmount": "190.00",
```

```
      "availableAmount": "10101.22"
```

```
    }
```

```
  }
```

```
}
```

- Update configuration URL

config/update

It is a POST request - endpoint is <https://api.yowpay.com/config/update>

- headers

X-App-Access-Ts - the timestamp when the call is initiated (UTC must be used)

X-App-Token - the **app token** (part of E-Commerce credentials explained above)

X-App-Access-Sig - the signature of the body (hash data). In order to receive you must use the **secret key** from credentials to hash the body content of the call

- body (as json)

- **returnUrl** (required)
- **cancelUrl** - (required)
- **webhookUrl** - (required)
- **timestamp** - the same timestamp that is set in the header (required)

Webhooks

As explained above, you must configure webhooks in order to activate this feature. For every business/ecommerce we have generated credentials (**app token** and **secret key**) and a webhook URL can be added. Based on the business/ecommerce configuration different types of webhooks can be sent. All type of webhooks are containing the following headers and body parameters

- In order to provide security of the **webhooks** we send the following **headers**:

- X-App-Access-Ts** - this is the same timestamp that is in the body as a parameter
- X-App-Token** - this is the **app token** (part of business credentials)
- X-App-Access-Sig** - this is the signature of the webhook (hash data)
- Content-type: application/json**
- Idempotency-Key** - this is a unique id of the webhook in order to avoid processing it a second time (in case of network issue for example)

For creating the signature, the **secret key** from E-Commerce/Business credentials must be used. The secret key must not be exposed and it proves that data has not been manipulated.

```
hash_hmac('sha256', "BodyWithParameters", "YOUR_SECRET_KEY")
```

- **body** (as json)

timestamp - (this is the time of the webhook in UTC, it can be used as a security feature)
eventType - based on the value of the webhook type we have different parameters provided in the webhook body

The following actions can be done to prove the validity of the webhook.

- the body must be hashed with the secret key and the result must be equal to the hash/signature of the header
- the timestamp in the body and in the header must be equal
- the timestamp should be relatively fresh (not older than 30 seconds for example). Be careful that UTC is used to avoid confusion with different server times.

The **eventType** field can have the following values which determines the different webhook types:

- **transaction.credited** → The only webhook activated by default. Triggered when the funds are actually settled. This is the most used one and is active by default.
- **payment.status.updated** → Triggered when the customer is redirected to their bank, when the payment is confirmed, or when the payment is refused.
- **transaction.unreconciled** → Triggered when the funds are credited but the transaction cannot be linked to previously created payment request
- **refund.confirmed** → Triggered when the refund is confirmed
- **refund.rejected** → Triggered when the refund is rejected

Based on the type of the webhook, it contains different structure.

– **transaction.credited**

- **timestamp** (this is the time of the webhook in UTC, it can be used as a security feature)
- **eventType** (transaction.credited)
- **paymentRequestId** (the id of the payment request that was created originally - can be used in API call)
- **transactionId** (the id of the actual transaction - it can be used to fetch transaction data via API call)

- **amount** (!!! this is the amount of the payment request - it can be different from an actual payment !!! example - 12.34)
- **currency** (!!! this is the currency of the payment request - it can be different from an actual payment !!! 3 letters code)
- **amountPaid** - (!!! this is the amount of the actual funds received - it can be different from the amount of the original payment request!!! example - 12.34)
- **currencyPaid** - (!!! this is the currency of the actual funds received - it can be different from the currency of the original payment request!!! 3 letters)
- **language**
- **reference** (YowPay reference)
- **orderId**
- **createDate** (payment request creation time - format ISO-8601 - e.g. 2025-02-15T10:40:24Z)
- **validateDate** (transaction validation time - format ISO-8601 - e.g. 2025-02-15T10:40:24Z)
- **senderIban**
- **senderSwift**
- **senderAccountHolder** - (account holder name of the sender)
- **status** (**1** for payment **credited** with expected amount, **2** for payment **credited** but **amount and currency don't match** with the original payment request)
- **paymentReference** (text that should appear on bank statement)
- **bankReference**

The **status** field can be equal to **1** or **2**.

1 when the payment is validated and the paid amount is matching to transaction amount

2 when the payment is validated but the amount and currency, that were paid, are not matching to amount and currency of the transaction. That is why we have also fields **amountPaid** and **currencyPaid** that are showing what amount was actually validated

Example:

```
{
  "timestamp": 1757585483,
  "eventType": "transaction.credited",
  "paymentRequestId": 174086,
  "transactionId": 2740186,
  "amount": "69.15",
  "currency": "EUR",
  "language": "",
  "reference": "YPJAGFKK",
  "orderId": "BILLID_11352038",
  "createDate": "2025-03-21T12:35:49+00:00",
  "validateDate": "2025-03-26T17:05:05+00:00",
```

```

"senderIban": "BE74977104862707",

"senderSwift": "PAYVBEB2XXX",

"senderAccountHolder": "Mayert, Wintheiser and Hegman",

"status": 1,

"amountPaid": "69.15",

"currencyPaid": "EUR",

"bankReference": "edf4db91-8646-4b51-8fd6-b2738fe0fc2b",

"paymentReference": "text on statement"
}

```

– transaction.unreconciled

- **timestamp** (this is the time of the webhook in UTC, it can be used as a security feature)
- **eventType** (transaction.unreconciled)
- **transactionId** (the id of the actual transaction - it can be used to fetch transaction data via API call)
- **amountPaid** - (this is the amount of the actual funds received, example - 12.34)
- **currencyPaid** - (this is the currency of the actual funds received, 3 letters)
- **reference** (text that should appear on bank statement)
- **createDate** (payment request creation time - format ISO-8601 - e.g. 2025-02-15T10:40:24Z)
- **validateDate** (transaction validation time - format ISO-8601 - e.g. 2025-02-15T10:40:24Z)
- **senderIban**
- **senderSwift**
- **senderAccountHolder** - (account holder name of the sender)
- **bankReference**

Example:

```

{

"timestamp": 1757585483,

"eventType": "transaction.unreconciled",

"transactionId": 2740186,

"reference": "text on statement",

"createDate": "2025-03-21T12:35:49+00:00",

"validateDate": "2025-03-26T17:05:05+00:00",

"senderIban": "BE74977104862707",

"senderSwift": "PAYVBEB2XXX",

```

```
"senderAccountHolder": "Mayert, Wintheiser and Hegman",  
"amountPaid": "69.15",  
"currencyPaid": "EUR",  
"bankReference": "edf4db91-8646-4b51-8fd6-b2738fe0fc2b"  
}
```

– payment.status.updated

- **timestamp** (this is the time of the webhook in UTC, it can be used as a security feature)
- **eventType** (payment.status.updated)
- **paymentRequestId** (the id of the payment request that was created originally - can be used in API call)
- **amount** (this is the amount of the payment request, example - 12.34)
- **currency** (this is the currency of the payment request, 3 letters code)
- **language**
- **reference** (YowPay reference)
- **createDate** (payment request creation time - format ISO-8601 - e.g. 2025-02-15T10:40:24Z)
- **orderId**
- **paymentInitiationStatus** (**1** for payment **credited** with expected amount, **2** for payment **credited** but **amount and currency don't match** with the original payment request)

The **paymentInitiationStatus** field (applies only to PIS transactions, when eventType = payment.status.updated) can have the following values

1 Payment initiation created (customer redirected to bank)

2 Payment initiation executed (validated by the client)

3 Payment initiation rejected

!! **paymentInitiationStatus** = 2 doesn't mean that the funds are credited, only **eventType** = transaction.credited, !!

Example:

```
{  
"timestamp": 1757585483,  
"eventType": "payment.status.update",  
"paymentRequestId": 2740186,  
"amount": "69.15",  
"currency": "EUR",  
"language": "",  
"reference": "YPJAGFKK",
```

```
"orderId": "BILLID_11352038",  
"createDate": "2025-03-21T12:35:49+00:00",  
"paymentInitiationstatus": 2  
}
```

– refund.confirmed

- **timestamp** (this is the time of the webhook in UTC, it can be used as a security feature)
- **eventType** (refund.confirmed)
- **originalTransactionId** (the id of the transaction that was refunded)
- **transactionId** (the id of the actual transaction - it can be used to fetch transaction data via API call)
- **status** (1 for confirmed refund)
- **amount** - (the amount of the refund, example - 12.34)
- **currency** -(the currency of the refund, 3 letters)
- **createDate** (payment request creation time - format ISO-8601 - e.g. 2025-02-15T10:40:24Z)
- **actionDate** (transaction validation time - format ISO-8601 - e.g. 2025-02-15T10:40:24Z)
- **beneficiaryIban**
- **beneficiarySwift**
- **beneficiaryAccountHolder** - (account holder name of the recipient)
- **paymentReference** (text that should appear on bank statement)
- **bankReference**

Example:

```
{  
  "timestamp": 1757585483,  
  "eventType": "refund.confirmed",  
  "status": 1,  
  "originalTransactionId": 765432,  
  "transactionId": 2740186,  
  "createDate": "2025-03-21T12:35:49+00:00",  
  "actionDate": "2025-03-26T17:05:05+00:00",  
  "beneficiaryIban": "BE74977104862707",  
  "beneficiarySwift": "PAYVBEB2XXX",  
  "beneficiaryAccountHolder": "Mayert, Wintheiser and Hegman",  
  "amount": "69.15",  
  "currency": "EUR",  
}
```

```
"bankReference": "edf4db91-8646-4b51-8fd6-b2738fe0fc2b",
```

```
"paymentReference": "text on statement"
```

```
}
```

– refund.rejected

- **timestamp** (this is the time of the webhook in UTC, it can be used as a security feature)
- **eventType** (refund.rejected)
- **originalTransactionId** (the id of the transaction that was tried to be refunded)
- **transactionId** (the id of the actual transaction - it can be used to fetch transaction data via API call)
- **status** (9 for rejected refund)
- **amount** - (the amount of the refund, example - 12.34)
- **currency** -(the currency of the refund, 3 letters)
- **createDate** (payment request creation time - format ISO-8601 - e.g. 2025-02-15T10:40:24Z)
- **actionDate** (transaction rejection time - format ISO-8601 - e.g. 2025-02-15T10:40:24Z)
- **beneficiaryIban**
- **beneficiarySwift**
- **beneficiaryAccountHolder** - (account holder name of the recipient)
- **paymentReference** (text that should appear on bank statement)
- **bankReference**

Example:

```
{
```

```
"timestamp": 1757585483,
```

```
"eventType": "refund.rejected",
```

```
"status": 9,
```

```
"originalTransactionId": 765432,
```

```
"transactionId": 2740186,
```

```
"createDate": "2025-03-21T12:35:49+00:00",
```

```
"actionDate": "2025-03-26T17:05:05+00:00",
```

```
"beneficiaryIban": "BE74977104862707",
```

```
"beneficiarySwift": "PAYVBEB2XXX",
```

```
"beneficiaryAccountHolder": "Mayert, Wintheiser and Hegman",
```

```
"amount": "69.15",
```

```
"currency": "EUR",
```

```
"bankReference": "edf4db91-8646-4b51-8fd6-b2738fe0fc2b",
```

```
"paymentReference": "text on statement"
```

```
}
```

We have an option with several retries of the webhooks in case of failure

We consider a webhook successfully accepted when a **http code 200** is returned and the body content is a string message `"ok"` or json reply `{"result":"ok"}`.

Useful test cases

When testing the payment process we recommend to test also cases that look like not normal but they are real and there is a big probability to happen from time to time.

- Payment is done but there is no reference provided by the merchant. In that case we send a webhook for transaction unreconciled.

in this case merchant can check if previous payments were done with the same sender's IBAN, if yes then it is possible to automate the reconciliation of the unreconciled transaction based on the sender's IBAN.

- Payment is done but the expected amount from the payment request is not matching with the paid amount. In that case we send a webhook for transaction credited but we set the transaction as a mismatch. We provide also expected amount and currency and also transaction amount and currency.

In this case it is still possible to reconcile, depending on the business model the reference is passed through so the link to the end user in the ecommerce platform exists, just a different amount should be credited or an alert sent if the payment is not full.

- A second payment is done for already paid payment request. A user can repeat the payment and we send a new webhook for the new credited transaction. The duplicated payment can also have a different amount.

In this case a new payment request is created, with a new paymentRequestid, as the payment reference is the same there are no issues to credit the end user which is linked to the payment reference.

Walk-around reconciliation usual issues

First of all there should not be any “missing” transactions, we are sending web-hooks with retry queues, if there is an error on the merchant’s side, a timeout or better a 500 error should be returned, http 200 should be returned only when the treatment on the posted data was managed successfully.

In case of any doubt, you can check transactions credited and account balance via the API or via the portal. We advice to do daily checks to be sure that no transaction is missing.

Reports, statements, invoices are available via our portal.