Rabobank
COO Operations NL

Rabo OmniKassa 2.0 API

Developer's manual versie 1.6 UK

Version:	1.6 UK December 2018
Contact e-mail address:	contact@omnikassa.rabobank.nl

Inhoudsopgave

1.	*Introduction*	*1*
1.1.	*Overview*	*1*
1.2.	*Note on ' Expect: 100-Continue ' header*	*1*
2.	*Call: Refresh*	*2*
3.	*Call: Order Announce*	*4*
4.	*Call: Notification*	*23*
5.	*Call: Status pull*	*26*

1. Introduction

This document describes the Rabo Omnikassa API. This API can be used to connect a webshop to Rabo OmniKassa. In this document Rabo OmniKassa will be further abbreviated as ROK, but this abbreviation should not be used in communication or documentation of the plugin to Webshops.

The examples contain many long texts, such as tokens, signatures and URLs. To keep the text readable, these are provided with (extra) end-of-lines. Extra end-of-lines should not be stopped in the messages and will not be included in the real messages.

1.1. Overview

The payment interface consists of 3 calls from the web shop to Rabo OmniKassa (ROK) and 1 call from ROK to the web store.

The payment of an order at ROK consists of the following steps:

1. The web store requests an Access Token on ROK (Refresh call)
2. The web store uses the Access Token to announce an order (Order announce call)
3. The webshop leads the consumer to ROK
4. The consumer executes the payment using ROK
5. The consumer is guided back to the web store
6. ROK sends the web shop a message to indicate that an order status update is available (Notification call)
7. The webshop asks ROK the latest order statuses (Status pull call)

The calls are:

- Refresh
- Order announce
- Notification
- Status pull

In the calls tokens are used, these are JSON Web Tokens. It is not necessary for the webshop to be able to create or parse these tokens, all tokens are provided by Rabobank and ROK.

1.2. Note on 'Expect: 100-Continue' header

POST requests sent to ROK may not contain a header ' Expect: 100-Continuous '. This header is not supported by ROK and leads to a significant delay in processing time. This limitation is particularly relevant for (but not limited to) webshops that use CURL to send requests.

2. Call: Refresh

To be able to communicate with ROK in a safe manner, the webshop uses an **access token**. An access token has a limited validity (a number of hours). If the webshop no longer has a valid **access token**, then the online store must use its **refresh token** to retrieve a new access token. This access token can then be used for the next order announce calls. The webshop must keep track of how long an access token is valid, and when a new access token has to be retrieved.

The web store must cache the access token as long as it is still valid. It is not the intention that a new access token is collected for every payment if this is not necessary.

Request

Call

The call is an HTTPS GET request without body.

URL

Sandbox environment: *https://betalen.rabobank.nl/omnikassa-api-sandbox/gatekeeper/refresh*

Production environmnent: *https://betalen.rabobank.nl/omnikassa-api/gatekeeper/refresh*

Headers

The entry requires at least the entry:

```
Authorization: Bearer <refresh-token>
```

Where <refresh-token> is replaced by the refresh token as it can be copied from the Rabobank Dashboard.

Note: the *Authorization* Header is the only header to be added to the request to be able to communicate with the Omnikassa.

Example

```
Authorization: Bearer
eyJraWQiOiIrcUNTdzlvL2dGcUMxeVlHWVhHZFBReGFVVTVLYlpPYWVCZjNiOHFrYWxvPS
IsImFsZyI6IkVTMjU2In0.eyJta2lkIjoxMDI0LCJleHAiOjE0OTExNDU3MjN9.MEYCIQD
yt1p3lA3zTyl_s8nQuLwBburgW1w3pIrFk4FDyRXebAIhALnCHKUwc81RA7DAv_IF8uOtb
VcIzJVUc447I14KVe04
```

Response

Example response:

```
{
 "token":
"eyJraWQiOiJHS0wiLCJhbGciOiJFUzI1NiJ9.eyJwayMiOjEwMjQsImNpZCI6ImNjYTUt
NmRjYyIsImV4cCI6MTQ4MDAwNjQ5MX0.MEQCIGZoLp7HvBS6SbHVfwCICQz_jvF-
abvbET2HsENcAKG_AiAYlI8WouBYYRvkbJHamR_PBL36P1b2fCy2H5mZNQtS9Q",
 "validUntil": "2016-11-24T16:54:51.216+0000",
 "durationInMillis": 28800000
}
```


The content of the token field is the access token.

3. Call: Order Announce

An order is announced at ROK by an order announcement. This contains the details of the order. The result of this is a redirect URL where the browser of the consumer has to be sent.

Request

Call

The call is an HTTPS POST request with a JSON body.

URL

Sandbox environment: *https://betalen.rabobank.nl/omnikassa-api-sandbox/order/server/api/order*

Production environment: *https://betalen.rabobank.nl/omnikassa-api/order/server/api/order*

Headers

A minimum of two entries are required in the header:

```
Content-Type: application/json
Authorization: Bearer <access-token>
```

Where <access-token> is replaced by the access token recovered in the Refresh call.

Example:

```
Authorization: Bearer
eyJraWQiOiJHS0wiLCJhbGciOiJFUzI1NiJ9.eyJwayMiOjUwMiwiY2lkIjoiYzhjYy1mM
ThjIiwiZXhwIjoxNDc5MTIyODc2fQ.MEUCIQC2Z5WUVTAKcBHISsOVMJIJE8PAbVe5x1io
r4bgrTcgCwIgLNoVIWEmSbQekJTccM89sosAY-8JzN47DGjvdPGdF0w
```

Body

The details of the order are placed in the body as a JSON message. Not as a key value pair but as bare content of the body. The following fields are supported:

JSON structuur

Field	Number	Format	Explanation and remarks	Example

timestamp	1	DT	ISO 8601 standard Date / time on which the order is announced at ROK. As a rule, this is the current date / time. This field is mandatory and provides protection against so-called replay (playback) attacks	2017-02-06T08:32:51.759+01:00
merchantOrderId	1	AN (Strictly).. Max 24	Generated by Merchant. If you want your webshop to use AfterPay, this field must be unique. If the ID contains more than 24 characters, the extra characters are removed after the 24th character.	order123
description	0..1	AN..max 35	Description of the order. If the description contains more than 35 characters then the extra characters are removed after the 35th character.	A nice blue beach ball.
orderItems	0..1	Array van OrderItems	The orderlines, see below for more details.	

amount	1	Money	The total order amount in cents, including VAT. The amount must be equal to the sum over all order items of the piece price (including VAT) multiplied by the number of copies. As a result of the way in which the VAT is calculated, due to rounding differences it can occur that these are not the same. We therefore recommend to base the total VAT amount on the VAT of the piece price instead of the total order amount excluding VAT. See example **Note**: If the amount is not equal to the sum of the amounts of the order items then 1. the order items from the order announcement are filtered, and 2. AfterPay is not possible as a payment method	Suppose that the piece price of an order item (excluding VAT) is € 12.98 and a VAT rate of 21% is applied. When a consumer orders 7 copies, the piece price including VAT € 12.98 + 21% = € 15.71 is rounded off. The total order amount (including VAT) that is given in this field is then 7 x € 15.71 = € 109.97
shippingDetail	0..1	Address	The shippingadress, see below for more details.	
billingDetail	0..1	Address	The billingadress, see below for more details..	
customerInformation	0..1	CustomerInformation	The customer details, see below for more details.	
language	0..1	AN..2	ISO 639-1 Standard. Not case sensitive. If language is not determined then NL is automatically selected by the ROK API. The following languages are allowed: NL, EN, FR and DE.	NL

merchantReturnURL	1	AN..max 1024	The URL to which the consumer's browser will be sent after the payment.	https://mijn.webwinkel/betalingsresultaat
paymentBrand	0..1	AN..50	This field is optional and is used to enforce a specific payment method with the consumer instead of the consumer selecting a payment method on the payment method selection page. Valid values are: • IDEAL • AFTERPAY • PAYPAL • MASTERCARD • VISA • BANCONTACT • MAESTRO • V_PAY • CARDS The CARDS value ensures that the consumer can choose between payment methods: MASTERCARD, VISA, BANCONTACT, MAESTRO and V_PAY	IDEAL

paymentBrandForce	0..1	AN..50	This field should only be delivered if the paymentBrand field (see above) is also specified. This field can be used to send or, after a failed payment, the consumer can or can not select another payment method to still pay the payment. Valid values are: • FORCE_ONCE • FORCE_ALWAYS In the case of FORCE_ONCE, the indicated paymentBrand is only enforced on the first transaction. If this fails, the consumer can still choose another payment method. When FORCE_ALWAYS is chosen, the consumer can not choose another payment method.	FORCE_ONCE
signature	1	AN..128	Signature on the message, see heading Signature for more details.	1795e71284f9e3e5805fe43964259f28810cd5047a726a5aaa48043eac69e8728ce23e97032f4a0844b0509dbe9f247a2c99f06017fdb7b85b06710a1c666a

OrderItems

Field	**Number**	**Format**	**Explanation and remarks**	**Example**
id	0..1	AN..max 25	Item ID. If the ID contains more than 25 characters, the extra characters will be removed after the 25th character.	A1000
name	1	AN..max 50	Item name	Jackie O Round Sunglasses

description	0..1	AN..max 100	Item description. If the description contains more than 100 characters, the extra characters are removed after the 100th character.	These distinct, feminine frames balance a classic Jackie-O styling with a modern look.
quantity	1	N	number: 1-2147483647.	1
amount	1	Money	The amount in cents, including VAT, of the item each, see below for more details.	If the piece price of an order item (excluding VAT) is € 12.98 and a VAT rate of 21% is applied. The piece price including VAT € 12.98 + 21% = € 15.71
tax	0..1	Money	The VAT of the item each, see below for more details.	
category	1	AN..max 8	Product Category: PHYSICAL or DIGITAL. If the category is not (well) defined, then PHYSICAL selected is by the ROK API.	PHYSICAL
vatCategory	0..1	N	The VAT category of the product. The values refer to the different rates used in the Netherlands: 1 = high (currently 21%), 2 = Low (until 31 December 2018 this is 6%, then 9%), 3 = zero (0%), 4 = none (exempt from VAT)	1

Money

Field	Number	Format	Explanation and remarks	Example
amount	1	N	The amount in cents. The amount can be a negative number if a discount exists.	1000 or -1000 in case of discount.

currency	1	AN..max	The currency. ROK currently only supports EUR	EUR

Address

Field	Number	Format	Explanation and remarks	Example
firstName	0..1	AN..max 50	First name. Important. If the FirstName contains more than 50 characters then the extra characters are cleared away after the 50th character.	Jan
middleName	0..1	AN..max 20	Insert or second name. If the insert contains more than 20 characters then the additional characters are cleared away after the 20th character.	van
lastName	1	AN..max 50	Surname. If the surname contains more than 50 characters then the additional characters are cleared away after the 50th character.	Jansen
street	1	AN..max 100	Street. If the street contains more than 100 characters, the extra characters are removed after the 100th character. **Note**: In case of payment with **Visa**, **Mastercard**, **Bancontact**, **V PAY** and **Maestro** The street name is abbreviated to 50 characters.Street.	Beukenlaan

houseNumber	0..1	AN..max 100	House number. If the house number contains more than 100 characters, the extra characters are removed after the 100th character. **Note**: In case of payment with **Visa**, **Mastercard**, **Bancontact**, **V PAY** and **Maestro** This field and houseNumberAddition (see below) are merged and abbreviated to 10 characters.	12
houseNumber Addition	0..1	AN..max 6	House number additions. If the house number addition contains more than 6 characters then the extra characters are removed after the 6th character.	a
postalCode	1	AN..max 10	Postal code	1234AA
city	1	AN..max 40	City. If the city contains more than 40 characters then the additional characters are removed after the 40th character.	Amsterdam
countryCode	1	A..2	Country code, ISO 3166-1 alpha-2	NL

CustomerInformation

Field	**Number**	**Format**	**Explanation and remarks**	**Example**
emailAddress	0..1	AN..max 45	The e-mail address of the consumer. If the email address contains more than 25 characters, the extra characters will be removed after the 25th character.	jan@example.org
dateOfBirth	0..1	DD-MM-YYYY	The date of birth of the consumer	21-11-1977
gender	0..1	M/F	The gender of the consumer	F

initials	0..1	A..max 256	The initials of the consumer. If the initials contain more than 256 characters then the extra characters are removed after the 256ste character.	J.A.N.
telephoneNumber	0..1	AN..max 31	The consumer's telephone number If the phone number contains more than 31 characters, the extra characters are removed after the 31st character.	+31204971111

Fields formats

Format	Explanation	Example
A.. Max nn	A field that consists of letters and other characters, such as ".", "@", etc. This field contains a maximum of nn characters. HTML tags are not allowed.	J.A.N.
N	A field with this size consists of only numbers	234
AN (Strictly).. Max nn	A field that consists strictly of alphanumeric characters of Max nn characters	Order123
AN.. Max nn	A field that consists of alphanumeric and other characters, such as ".", "@", etc. This field contains a maximum of nn characters. HTML tags are not allowed.	jan@example.org
Dt	ISO 8601 date format.	2017-02-06T08:32:51.759 + 01:00
DD-MM-YYYY	Date format used at the date of birth of the consumer.	25-11-2000
M/F	Consumer sex.	F

An example of a minimum JSON message:

```
{
  "Timestamp": "2017-02-06T08:32:51.759 + 01:00",
```

```
  "MerchantOrderId": "Order123",
  "Amount": {
    "Currency" means "EUR",
    "Amount": "4999"
  },
  "MerchantReturnURL": "http://www.example.org",
  "Signature":
"1795e71284f9e3e5805fe43964259f28810cd5047a726a5aaa48043eac69e87e28ce2
3e97032f4a08444b0509dbe9f247a2c99f06017fdb7b85b06710a1c666a"
}
```

An example of a complete message:

```
{
  "timestamp": "2017-09-11T14:54:57+02:00",
  "merchantOrderId": "order123",
  "description": "Aankoop mijn webwinkel ordernummer 123",
  "orderItems": [
    {
      "id": "A1000",
      "name": "Jackie O Round Sunglasses",
      "description": "These distinct, feminine frames balance a
classic Jackie-O styling with a modern look.",
      "quantity": 1,
      "amount": {
        "currency": "EUR",
        "amount": 22500
      },
      "tax": {
        "currency": "EUR",
        "amount": 4725
      },
      "category": "PHYSICAL",
      "vatCategory": "1"
    }
  ],
  "amount": {
    "currency": "EUR",
    "amount": 22500
  },
  "shippingDetail": {
    "firstName": "Jan",
    "middleName": "van",
    "lastName": "Jansen",
    "street": "Beukenlaan",
    "houseNumber": "12",
    "houseNumberAddition": "a",
    "postalCode": "1234AA",
    "city": "Amsterdam",
    "countryCode": "NL"
  },
  "billingDetail": {
    "firstName": "Jan",
    "middleName": "van",
    "lastName": "Jansen",
    "street": "Kersenstraat",
    "houseNumber": "385",
    "houseNumberAddition": "b",
    "postalCode": "1234BB",
```

```
    "city": "Haarlem",
    "countryCode": "NL"
  },
  "customerInformation": {
    "emailAddress": "jan@example.org",
    "dateOfBirth": "21-11-1977",
    "gender": "M",
    "initials": "J.A.N.",
    "telephoneNumber": "+31204971111"
  },
  "language": "nl",
  "merchantReturnURL": "https://mijn.webwinkel.nl/betalingsresultaat",
  "paymentBrand": "IDEAL",
  "paymentBrandForce": "FORCE_ONCE",
  "signature":
"1a0f6e4526fb81d5bf70fe2a9d7a3c46a2a1e85993b5df08cd2a7ff5324d48da6a12f
b38a38ccb4c4d44b9901f7f3a5b5fe6aec475db6fe24c57fcc4f2d6eff2"
}
```


Signature

The signature is the result of applying HMAC-SHA512 (initialized with the signing key) on the message.

The signing key can be copied from the Rabobank Dashboard. This is a shared secret between Rabobank and the web store. This allows Rabobank to check that it is the web store that sends the message. The key is delivered in base64 encoded form. When initializing the HMAC-SHA512, it must first be decoded base64.

Input

The signature is calculated on the following fields, in the same order as indicated, of the JSON object:

1. timestamp
2. merchantOrderId,
3. amount currency
4. amount amount
5. language**
6. description**
7. merchantReturnURL
8. (order items)*
9. (shipping detail)*
10. (paymentBrand)*
11. (paymentBrandForce)*
12. (customerInformation)*
13. (billing detail)*

The contents of these fields are combined together whilst being comma separated.

The (..) * fields are optional, if these fields do not have a value for a given order, they are not included in the payload for the signature calculation (also not as a blank value).

The * * fields are optional, but are always used as part of the signature. If they do not have a value for a given order, a blank string is included in the payload for each.

Examples:

The signature fields pasted together from the minimum sample message are:

```
2017-02-06T08:32:51.759 + 01:00, order123, EUR, 4999,,,
http://www.example.org
```

The signature fields that are pasted together from the full sample message are:

```
2017-09-11T16:28:07 + 02:00, order123, EUR, 22500, NL, purchase my
webshop order number 123,
https://mijn.webwinkel.nl/betalingsresultaat, A1000, Jackie O Round
sunglasses, These distinct, feminine frames balance a classic Jackie-O
styling with a Modern look., 1, EUR, 22500, EUR, 4725, PHYSICAL, 1,
Jan, van, Jansen, Beech Avenue, 12, A, 1234AA, Amsterdam, NL, IDEAL,
FORCE_ONCE, jan@example.org,21-11 -1977, M, J.A.N., +31204971111, Jan,
van, Jansen, Cherry Street, 385, B, 1234BB, Haarlem, NL
```

If the order announcement contains the following values:

timestamp = 2017-02-06T08:32:51.759 + 01:00
MerchantOrderId = order123
Language = en
Description = Order Descriptions
Currency = EUR
Amount = 4999
MerchantReturnURL = http://www.example.org

Then the payload for the calculation of the signature will look like this:

```
2017-02-06T08:32:51.759 + 01:00, order123, EUR, 4999, NL, description
order, http://www.example.org
```

In the example above, the timestamp-merchantReturnURL fields each have a value. These are therefore included in the payload. The (..) * in this example, fields do not have a value and are therefore not included in the payload.

If the order announcement contains the following values:

timestamp = 2017-02-06T08:32:51.759 + 01:00
MerchantOrderId = order123
Language =
Description =

Currency = EUR
Amount = 4999
MerchantReturnURL = http://www.example.org

Then the payload for the calculation of the signature will look like this:

```
2017-02-06T08:32:51.759 + 01:00, order123, EUR, 4999,,,
http://www.example.org
```

The Language and Description fields have no values. Therefore, for these fields, a blank string is included in the payload at position 5 and 6. The (..) * In this example, fields have no value again. These are therefore not included in the payload.

If the order announcement contains the following values:

timestamp = 2017-02-06T08:32:51.759 + 01:00
MerchantOrderId = order123
Language =
Description =
Currency = EUR
Amount = 4999
MerchantReturnURL = http://www.example.org
Order items =
ShippingDetail =
PaymentBrand = IDEAL
PaymentBrandForce = FORCE_ONCE

Then the payload for calculating the signature will look like this:

```
2017-02-06T08:32:51.759 + 01:00, order123, EUR, 4999,,,
http://www.example.org, IDEAL, FORCE_ONCE
```

In This example, the PaymentBrand and PaymentBrandForce fields have a value and are therefore added to the payload. Because the (..) * Fields CustomerInformation and billingDetail have no value these are not taken into account.

Order Items

The fields of the order items are added in the following order:

1. (id)*
2. name
3. description
4. quantity
5. amount currency
6. amount amount

7. (tax velden) **
8. category
9. (vatCategory)*

The fields are combined together in the same way as with the order. If a field is not present, the content is considered empty.
The (..) * fields are optional, if they are not given in the request then the Webshop application should not use these fields to calculate the signature.
* * Tax is an optional field, if it is not present, it is considered a single blank field. If it is present, the amount and currency are added as two separate fields.

Examples

If the order announcement contains the following values:

timestamp = 2017-02-06T08:32:51.759 + 01:00
MerchantOrderId = order123
Language = en
Description = Order Descriptions
Currency = EUR
Amount = 100
MerchantReturnURL = http://www.example.org
id = itemId123
Name = ItemName
Description = Item Details
Quantity = 2
Currency = EUR
Amount = 50
Currency (tax) = EUR
Amount (Tax) = 0
Category = PHYSICAL
VatCategory = 3

Then the payload for calculating the signature will look like this:

```
2017-02-06T08:32:51.759 + 01:00, order123, EUR, 100, NL, Description
order, http://www.example.org, itemId123, item name, description item,
2, EUR, 50, EUR, 0, PHYSICAL, 3
```

If the order announcement contains the following values:

timestamp = 2017-02-06T08:32:51.759 + 01:00
MerchantOrderId = order123
Language = en
Description = Order Descriptions
Currency = EUR
Amount = 100
MerchantReturnURL = http://www.example.org
ID =
Name = ItemName

Description = Item Details
Quantity = 2
Currency = EUR
Amount = 50
Currency (tax) =
Amount (tax) =
Category = PHYSICAL
VatCategory =

Then the payload for the calculation of the signature will look like this:

```
2017-02-06T08:32:51.759 + 01:00, order123, EUR, 100, NL, Description
order, http://www.example.org, item name, description item, 2, EUR,
50,,, PHYSICAL
```

In This example, the (..) * Currency (tax) and amount (tax) fields are not value. Therefore, an empty string is included in the payload for these fields (on positions 13 and 14).

Shipping & billing details

The fields of the shipping and billing details are added in the following order:

1. firstName
2. middleName
3. lastName
4. street
5. (houseNumber)*
6. (houseNumberAddition)*
7. postalCode
8. city
9. countryCode

The fields are combined together in the same way as with the order. If a field is not present, the content is considered empty.

The (..) * fields are optional, if they are not given in the request then the Web Store application should not use these fields to calculate the signature.

Customer information

The fields of the customer information are added in the following order:

1. emailAddress**
2. dateOfBirth**
3. gender**
4. initials**
5. telephoneNumber**

The * * fields are optional, but are always used as part of the signature. If they are not given in the request then the Webshop application must use a blank value as part input for the signature.

Calculation

To arrive at the signature an HMAC-SHA512 calculation is performed. The HMAC-SHA512 is initialized with the signing key from the dashboard. Note that this signing key is supplied in base64 encoded form and therefore base64 must be decoded first. Subsequently, the byte representation of the string of the stitched signature fields (based on UTF-8 encoding) is presented to the HMAC-SHA512 to arrive at the signature. The hexadecimal representation in lower case letters of this signature is finally included in the message.

Example

PHP

```
$signatureFields = '2017-02-
06T08:32:51.759+01:00,order123,EUR,4999,,,http://www.example.org';
$signingKey = '<signing key uit Rabobank Dashboard>';
$signingKeyDecoded = base64_decode($signingKey);

$signature = hash_hmac('sha512', $signatureFields,
$signingKeyDecoded);
```

Java

```
String signatureFields = "2017-02-
06T08:32:51.759+01:00,order123,EUR,4999,,,http://www.example.org";
String signingKey = "<signing key uit Rabobank Dashboard>";
byte[] signingKeyDecoded = Base64.getDecoder().decode(signingKey);


Mac macAlgorithm = Mac.getInstance("HmacSHA512");
Key secretKey = new SecretKeySpec(signingKeyDecoded , "HmacSHA512");
macAlgorithm.init(secretKey);

byte[] result = macAlgorithm.doFinal(signatureFields.getBytes(UTF_8));
String signature = Hex.encodeHexString(result);
```

C#

```
 String signatureFields = "2017-02-
06T08:32:51.759+01:00,order123,EUR,4999,,,http://www.example.org";
 String signingKey = "<signing key uit Rabobank Dashboard>";
 byte[] signingKeyDecoded = Convert.FromBase64String(signingKey);

 string signature = "";
 using (var hmacsha512 = new HMACSHA512(signingKeyDecoded))
 {
     var stream = new
MemoryStream(Encoding.UTF8.GetBytes(signatureFields));
     foreach (var b in hmacsha512.ComputeHash(stream))
     {
         signature = signature + $"{b:x2}";
     }
```

```
 }
```

Discount

A discount in the order is also provided by an order item but the amount and (if applicable) and tax fields have a negative value. The following examples set a discount of 10 euros:

Java

```
{
  "id": "1234",
  "Name": "Discount",
  "description": "One-time discount",
  "Quantity": 1,
  "Amount": {
    "Currency" means "EUR",
    "Amount":-1000
  },
  "Tax": {
    "Currency" means "EUR",
    "Amount":-210
  },
  "category": "PHYSICAL",
  "VatCategory": "1"
}
```

Response

The answer is a JSON object containing the redirect URL and a signature.

Example response:

```
{
 "signature":
"d3dd97b48752f3d4d4c5a914bf9e935956546887c7c8fd020a0702cd4462fbd8c60d2
b7b0e0c4fc160005c71a1f7c504ef7ca8bbfb82cf0a6564b1bfeb0a4f7f",
 "redirectUrl": "https://betalen.rabobank.nl/omnikassa-api/payment-
brand?token=eyJraWQiOiJFTU8iLCJhbGciOiJFUzI1NiJ9.eyJlbW8iOiJhYWZhMDAxM
y1lYmNiLTQ1ZjQtYTRmYi01OGNjMmQ5MDM2MDIiLCJjaWQiOiIxOTQwLTBkNTgiLCJleHA
iOjE0ODAxNTA0Mjd9.MEQCIHJLZjlcNYShX7YzVFvghfwmvH7WTV2Lj5IQIejFyjH7AiBK
mvahL29DgiA5vMhGLOHoHaT3SjQKgR4RVxJetG7Fdw&lang=nl"
}
```

The signature is calculated in the same way as the signature of the request, but in this case there is only one field: redirectUrl.

The signature in the response is calculated in the same way as the signature of the request for announcing the order, the difference is that in the response the signature is based on redirectUrl.

Required fields per payment method

Regardless of the payment method, at least the merchantOrderId, amount and the merchantReturnUrl must be included in each order. Depending on the payment method, additional information must be included in the order. The table below shows what data this is, broken down by payment method:

Payment method	Additional information in the order
iDEAL	No additional information is required for this payment method.
PayPal	Although not mandatory, we also recommend that you include order items in the order for additional security regarding PayPal's payment to the Webshop owner.
Bancontact Visa MasterCard V PAY Maestro	Although not mandatory, we advise you to include the delivery address (or, if not known, the billing address) in the order for additional certainty about a successful payment.
AfterPay	For AfterPay, the following additional information is required: • Order items with per order item the sales tax amount or the sales tax category. • Billing or delivery address (if different then both addresses are required). In addition, AfterPay requires that the MerchantOrderId field be unique. The order amount must be at least 5 euro.

If for a payment method The mandatory additional information is missing in the order then the consumer cannot use this method to fulfill the payment request. If the payment method was included in the order using the PaymentBrand field, the announcement will be refused by Rabo Omnikassa.

Consumer is redirected to the ROK hosted payment pages

The webshop must use the redirectUrl to redirect the consumer to the ROK payment page. On this page, the consumer can choose between different payment methods to complete the payment or the consumer can choose to go directly back to

Consumer returns to the webshop

When the consumer returns to the webshop, URL parameters are added to say something about the status of the order. Unfortunately, the final status of an order is not known in all cases, for example with an iDEAL payment it may take a while before the status is final.

The URL parameters are:

URL Parameter	**Meaning**	**Example**
order_id	The "merchantOrderId" as used in the Order announce.	order123
status	The status of the order, see below for more details.	COMPLETED
signature	The signature of the data in the URL.	14bf9e935956546887c7c8fd020a0702cd4462d3dd97b48752f3d4d4c5a9cf0afbd8c60d2b7b0e0c46564b1bfeb0a4f7ffc160005c71a1f7c504ef7ca8bbfb82

Statuses

COMPLETED
The payment was successful.

EXPIRED
The consumer has not paid within the stipulated period.

IN_PROGRESS
The payment has not yet been completed. This can occur as a result of a breakdown or delay in the hinterland of payment processing. This is a possible outcome of an iDEAL or credit card payment.

CANCELLED

The consumer chose not to pay.

Signature

The signature is calculated in the same way as other signatures. In this case, the two fields (in order: order_id, status) are used as input.

4. Call: Notification

A notification is a message from ROK to the web shop to announce that at least one order has reached a definitive status. The notification is an invitation to the webshop to retrieve the statuses of these orders (status pull call). The webshop is free to do this immediately (even before an answer has been given to the notification) or at a later time. The notification contains a token that must be used in the status pull call, and this token has a limited validity (minutes). If the webshop does not make a status update call, ROK will send a new notification later. However, ROK will give up after a number of hours until the next order is processed.

The notification is signed by ROK so that the webshop can check its integrity. Analogous to the other messages, the signature is calculated using the signing key as configured for the web store.

If several signing keys are active for the web shop at ROK, ROK will send a notification to the web store separately for each key. ROK does not know which of these signing keys is configured within the webshop. The web store can carry out a successful verification of the signature from exactly one notification. The authentication of this notification can be used by the webshop for the status pull call. ROK will sign the statuses of these orders with the same signing key.

Request

The call is an HTTPS POST with the relevant information in the body as a JSON object.

URL

ROK sends the notification message to the webhook URL as specified in the Rabobank Dashboard.

Headers

The header contains at least the entry:

```
Content-Type: application/json
```

Body

The body of the notification is a JSON object containing the following fields:

Field	Meaning	Example

authentication	The token that can be used to do the status pull	eyJraWQiOiJOTyIsImFsZyI6IkVTMjU2In0.eyJubyMiOjEyMywibWtpZCI6NSwibm8kIjoibWVyY2hhbnQub3JkZXIuc3RhdHVzLmNoYW5nZWQiLCJjaWQiOiJhYmNkLTEyMzQiLCJleHAiOjE0ODg0NjQ1MDN9.MEUCIHtPFoKmXAc7JNQjj0U5rWpl0zR9RsQvgj_n-ckHBngHAiEAmbtgrxaiy4cS3BTHd0DJ8md3Rn7V13Nv35m5DurY1wI
expiry	The validity period of the token, in the ISO-8601 format (yyyy-MM-ddTHH: mm: ss.SSSZZ)	2016-11-25T09:53:46.765+01:00
eventName	The type of notification. For the time being this is always:merchant.order.status.changed	merchant.order.status.changed
poiId	Identification of the webshop (point of interaction), seen from ROK. This is relevant if several webshops use the same webhook URL.	123
signature	The signature of the message, see heading signature for details.	

Signature

The signature on the notification message is calculated in the same way as the other signatures. The fields used are in order:

- authentication
- expiry
- eventName
- poiId

For the example, the signature is determined based on the following string:

```
eyJraWQiOiJOTyIsImFsZyI6IkVTMjU2In0.eyJubyMiOjEyMywibWtpZCI6NSwibm8kIjoibWVyY2hhbnQub3JkZXIuc3RhdHVzLmNoYW5nZWQiLCJjaWQiOiJhYmNkLTEyMzQiLCJleHAiOjE0ODg0NjQ1MDN9.MEUCIHtPFoKmXAc7JNQjj0U5rWpl0zR9RsQvgj_n-ckHBngHAiEAmbtgrxaiy4cS3BTHd0DJ8md3Rn7V13Nv35m5DurY1wI,2016-11-25T09:53:46.765+01:00,merchant.order.status.changed,123
```

Response

ROK does not expect a response to this message and has set a short time-out for its request. It is therefore possible that the connection is already broken when a response is sent.

The reason for this is that it is often useful to immediately make the status pull call before a response is sent.

5. Call: Status pull

With this call, the web store retrieves final statuses from orders.

Request

Call

The call is an HTTPS GET with empty body.

URL

Sandbox environment: *https://betalen.rabobank.nl/omnikassa-api-sandbox/order/server/api/events/results/merchant.order.status.changed*

Production environment: *https://betalen.rabobank.nl/omnikassa-api/order/server/api/events/results/merchant.order.status.changed*

Headers

The entry requires at least the entry:

```
Authorization: Bearer <notification-token>
```

The <notification-token> is the field "authentication" from the notification.

Response

The response body contains a JSON object with the following fields:

Field	Meaning	Example
moreOrderResultsAvailable	Indication if there are more order results available than in this message. In that case, a status pull call can be made (with the same notification token). This can be repeated until the result is false.	true
orderResults	An array containing the results per order	See heading OrderResults for details

signature	The signature of the message, see heading signature for details.	99ca2487243fbad72bbaa456a3219db7b0d2a19777f436cedb3c045e999b86c05001bb0837b43caa3d1757321d00959ac2a161f473a103af72bf440db5147b4a

orderResults velden

Fields	**Meaning**	**Example**
merchantOrderId	OrderId as delivered during the Order Announce	order123
omnikassaOrderId	The unique id that the omnikassa has assigned to this order	1d0a95f4-2589-439b-9562-c50aa19f9caf
poiId	Unique identification of the webshop (point of interaction), seen from ROK. This is relevant if several webshops use the same webhook URL.	2004
orderStatus	The status of the order. See chapter "Consumer returns at the webshop" for an overview of the possible statuses.	CANCELLED
orderStatusDateTime	The moment this status is reached.	2016-11-25T13:20:03.157+01:00
errorCode	Future field, for now: always empty	
paidAmount currency	The currency in which payment is made	EUR
paidAmount amount	The amount paid by the consumer	0
totalAmount currency	The currency of the order	EUR
totalAmount amount	The total amount of the order, in cents	4999

Example response


```
{
  "signature":
"99ca2487243fbad72bbaa456a3219db7b0d2a19777f436cedb3c045e999b86c05001b
b0837b43caa3d1757321d00959ac2a161f473a103af72bf440db5147b4a",
  "moreOrderResultsAvailable": false,
  "orderResults": [
```

```
    {
      "merchantOrderId": "order123",
      "omnikassaOrderId": "1d0a95f4-2589-439b-9562-c50aa19f9caf",
      "poiId": "2004",
      "orderStatus": "CANCELLED",
      "orderStatusDateTime": "2016-11-25T13:20:03.157+01:00",
      "errorCode": "",
      "paidAmount": {
        "currency": "EUR",
        "amount": "0"
      },
      "totalAmount": {
        "currency": "EUR",
        "amount": "4999"
      }
    }
  ]
}
```


Signature

The signature on the result is calculated in the same way as the other signatures. The fields used are in order:

- moreOrderResultsAvailable
- De orderResults

For every orderResult the following fields are used in order:

- merchantOrderId
- omnikassaOrderId
- poiId
- orderStatus
- orderStatusDateTime
- errorCode
- paidAmount currency
- paidAmount amount
- totalAmount currency
- totalAmount amount

For the example, the signature is determined based on the following string:

```
false,order123,1d0a95f4-2589-439b-9562-
c50aa19f9caf,2004,CANCELLED,2016-11-
25T13:20:03.157+01:00,,EUR,0,EUR,4999
```

.