Conto QuickPay Payment Initiation Module. Integration Guide

Intro

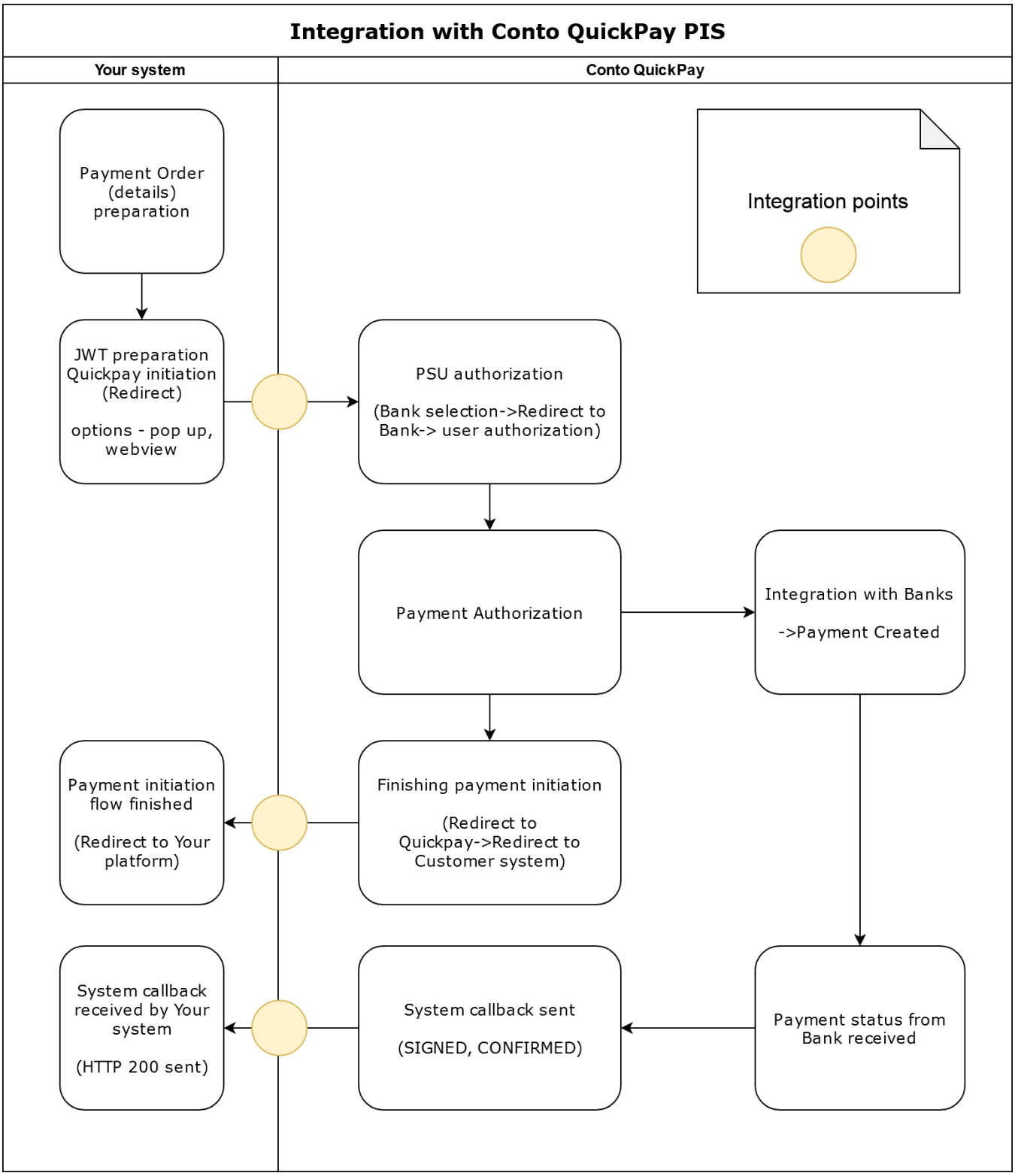
Welcome to Conto QuickPay payment initiation module integration guide! This document will assist you step by step in how to start using our payment initiation module. Just follow the instructions and get your system integrated with ours.

Plugins

There are ongoing developments for open source projects' plugins that allow seamless integration with our module. Let us know at support@contomobile.com what specific open source project you use.

Integration points

In fact, there are three basic integration points that have to be implemented for the integration to be complete. These integration points are marked by yellow circles in the diagram below:



1. JWT preparation and Conto QuickPay initiation.
   Here you have to create JWT to invoke a payment initiation window. The payment initiation window is invoked by JavaScript using our Conto QuickPay widget.
2. Payment initiation flow finished.
   On having finished the payment initiation flow, the user is redirected back to your system where you display a particular result based on the outcome.
3. System callback received by Your system.
   Eventually, a callback message, containing the actual status of the payment, is sent to your system.

Each of these integration points will be described in more detail in later chapters.

Payment initiation

The payment is initiated by JavaScript. Therefore, it is necessary to include the following script into your HTML:

HTML snippet

```
<script type="text/javascript" src="https://quickpay.contomobile.com/static/widget.js"></script>
```

This script will allow you to start the payment initiation window by creating a QuickPayWidget object with the parameters described in the table below.

Widget configuration

Widget initialization properties

Property	Explanation	Required	Type	Default
redirectUrlSuccess	This success redirect URL will be used to notify about the success flow status. If this property is defined, redirectUrlCancel must also be defined. When redirectUrlSuccess and redirectUrlCancel are defined, the redirect flow is enabled; otherwise, the popup flow is initiated.	YES when redirectUrlCancel is defined	string	–
redirectUrlCancel	This cancelled redirect URL will be used to notify about the cancelled flow status. If this property is defined, redirectUrlSuccess must be also defined. When redirectUrlSuccess and redirectUrlCancel are defined, the redirect flow is enabled; otherwise, the popup flow is initiated.	YES when redirectUrlSuccess is defined	string	-
defaultLanguage	This variable is used when there is a need to set a certain default language when the widget is initialized. The widget will allow the user to change the language in the widget as well. The language code should be provided in the ISO 639-2 code format.	NO	string	LIT (possible values – LIT, ENG)
defaultCountry	This property will allow setting a default country. The country code should be provided in the ISO 3166-1 Alpha-2 code format.	NO	string	LT (one possible value – LT)
creditor	Using this variable, you can pre-select a choice of a bank, i.e. if the customer specifies the desired bank to be directed to for making a payment, the system immediately shall direct the customer to the pre-selected bank (the creditor's value); otherwise (if such a value is not specified – NO by default), the entire list of available banks will be displayed to the customer. In order to be able to use this option, provide the BIC for each bank: Available Banks API. Value "OTHER" is used, if there is a need to initiate a payment with just providing the payment details.	NO	string	–
other	This variable adds/removes (true/false) option "Another bank" in the window of banks' selection. This chosen window provides only the payment details (redirection to the bank is not available).	NO	boolean	true (possible values – true, false)
webview	If you wish to integrate the payment initiation flow into your mobile application, set this property to "true". redirectUrlSuccesss and redirectUrlCancel must be defined.	NO	boolean	false (possible values – true, false)

Widget initialization example:

Widget initialization

```
        var host = 'https://quickpay.contomobile.com/';
new QuickPayWidget(host, token, {
    redirectUrlSuccess: 'https://yourshop.com/success_redirect_flow',
                redirectUrlCancel: 'https://yourshop.com/canceled_redirect_flow',
    defaultLanguage: 'LIT',
    defaultCountry: 'LT',
    creditor: 'UAAMLT21XXX',
    other: true,
    webview: false,
}).onSuccess(function () {
    // add your logic on success
}).onCancel(function () {
    // add your logic on cancel
}).initialize();
```

Widget sandbox initialization example:

Widget initialization

```
        var host = 'https://pis-sandbox.contomobiledev.com';
new QuickPayWidget(host, token, {
    redirectUrlSuccess: 'https://yourshop.com/success_redirect_flow',
                redirectUrlCancel: 'https://yourshop.com/canceled_redirect_flow',
    defaultLanguage: 'LIT',
    defaultCountry: 'LT',
    creditor: 'UAAMLT21XXX',
    other: true,
    webview: false,
}).onSuccess(function () {
    // add your logic on success
}).onCancel(function () {
    // add your logic on cancel
}).initialize();
```

Host

Host is a static value that should be https://quickpay.contomobile.com/. It is required for the widget to know how to reach our system.

Token

At initiating the payment window, you need to submit a *token*, which is JWT token. Token is signed using the HS256 algorithm.

> What is JWT?
>
> JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. This information can be verified and trusted because it is digitally signed.
>
> It consists of three main parts encoded in base64 and dot-separated: header.payload.signature
>
> For more information refer to: https://jwt.io/

JWT contains the details of the payment and the information that lets us know that this is you.

JWT Header is a constant and does not change:

JWT Header

Property	Explanation	Required	Type	Restrictions
alg	The algorithm used for the signature.	YES	string	The value must be "HS256"
typ	The type of the token.	YES	string	The value must be "JWT"

JWT Payload contains the details about the payment and you:

JWT Payload				
Property	**Explanation**	**Required**	**Type**	**Restrictions**
contractId	This property allows us identifying you, and we shall provided it to you at the beginning of the integration.	YES	string	Max length – 140 characters
amount	The payment initiation amount. Note that the payment is only in the EUR currency.	YES	number	Min value – “0.01”
paymentPurpose	The purpose of the SEPA payment.	YES	string	Max length – 140 characters
transactionId	The transaction ID provided by you. It should be unique for each payment initiation. It allows you to correlate the transaction with its status when the payment status callback is received.	YES	Max36 Text	Max length – 36 characters
beneficiaryAccount	The beneficiary account IBAN to be used instead of the default beneficiary account IBAN defined in the contract. This parameter is accepted only if the beneficiary account is registered in the system and multiple beneficiary account support is enabled for the contract.	NO	string	Max length – 34 characters
exp	The time when JWT expires.	NO	long	

Example

Let us assume that you need to make a 10 euro payment with the payment purpose set as "MIP test payment", then JWT signed with a secret `Bzg5Js61a0JUXmNT3q2qwoe2YEGuUtxStRILbDEe0zg=` would look like this:

JWT

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJjb250cmFjdElkIjoiNzE0NDYyIiwiYW1vdW50IjoxMCwicGF5bWVudFB1cnBvc2UiOiJNSVAgdGVzdCBwYXltZW50IiwidHJhbnNhY3Rpb25JZCI6ImRjMjgyMTZiZTVkMDQyMjg4YWE5MzU4NGIyMmVjMmE5In0.a-jIP5wuAluCjJRbKQcDcugXwQlS2P2801tOFr8SUwQ
```

JWT

JWT header

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

JWT payload

```
{"contractId":"714462","amount":10,"paymentPurpose":"MIP test payment","transactionId":"
dc28216be5d042288aa93584b22ec2a9"}
```

redirectUrlSuccess and redirectUrlCancel (payment initiation flow finished)

After having finished the payment initiation flow, the user is redirected to the URL specified in the widget initiation.

Widget initiation example:

Widget initialization

```
            var host = 'https://quickpay.contomobile.com/';
    new QuickPayWidget(host, token, {
        redirectUrlSuccess: 'https://yourshop.com/complete',
                    redirectUrlCancel: 'https://yourshop.com/cancel'
    }).initialize();
```

If the user closes the widget during the payment initiation flow, then he/she will be redirected to https://yourshop.com/cancel. If the payment initiation flow was successfully completed, then the user is redirected to https://yourshop.com/complete.

This allows you to return the user back to your system and display the appropriate result, however this does not mean that the payment was truly successful. The actual result of the payment will be sent as a callback.

Popup

It is also possible to make the widget open in a separate popup window, in this case the user never leaves your system for the payment initiation. This happens when redirectUrlSuccess and redirectUrlCancel are not defined.

Example:

Widget initialization

```
        var host = 'https://quickpay.contomobile.com/';
new QuickPayWidget(host, token, {
}).onSuccess(function () {
    // add your logic on success
}).onCancel(function () {
    // add your logic on cancel
}).initialize();
```

onSuccess() and onCancel() callbacks will be used with the popup flow. OnSuccess() is called when the payment flow has been successfully finished; onCancel() means that the user has cancelled the payment flow.

Creditor Bank

It is possible to choose a bank at your website before starting the payment initialization. Then each bank would have its own widget initialization script that looks like this:

Widget initialization

```
        var host = 'https://quickpay.contomobile.com/';
new QuickPayWidget(host, token, {
    redirectUrlSuccess: 'https://yourshop.com/complete',
                redirectUrlCancel: 'https://yourshop.com/cancel',
    creditor: "HABALT22"
}).initialize();
```

The bank is identified by BIC and it should be set to a creditor property. In this particular example, the creditor is "HABALT22".

Direct widget call not using widget.js

In order to call widget directly - via link - it is needed to configure QuickPayWidget url with its query parameters. All possible widget query parameters are listed in the *Widget initialization properties* table above. It is worth to mention, that there are two mandatory parameters:

- *token* - payment JWT token (which is explained above in Token paragraph)
- *webview* - should be set to *true* in order to get redirect flow without redirect urls to your shop

Query parameters values should be encoded. Widget initialization via link example:

Direct widget call

```
https://quickpay.contomobile.com/initialize?token=payment_JWT_token&redirectUrlSuccess=https%3A%2F%2Fquickpay.
contomobile.com%2Fconclusion&webview=true&
```

System Callback

A system callback request is sent to your system informing you about the status of the payment. No system callback will be sent if the user closes the widget or the browser before he/she is requested to authorize the payment.

Payment status SIGNED is returned, if the user has successfully confirmed the payment; or status FAILED is returned, if the payment was not signed.

An additional confirmation callback with status CONFIRMED is available in case the beneficiary account is Contomobile account. It informs that funds have been received. The callback is sent only after the money has been booked to your account, if this process does not take too long, then the first callback with status SIGNED is skipped.

Status EXPIRED may be returned, if the payment status is not conclusive (FAILED, SIGNED, CONFIRMED) for a long period of time. This can happen when the user closes the browser; in such a case, the bank of the user might not change the payment status from pending to a more conclusive.

Transaction status	
Status	**Description**
CONFIRMED	The payment has been successfully signed and the funds were received.
SIGNED	The payment has been successfully signed.
FAILED	The payment has been rejected or user-cancelled.
EXPIRED	The payment status is not conclusive (CONFIRMED, SIGNED, FAILED) after a long period of time.

Request Callback

A request callback is an HTTP POST request with a JSON body that contains JWT with a payment status. JWT is signed with the same secret provided by us that was used for payment initialization. Token is signed using the HS256 algorithm.

Request JSON body			
Property	**Explanation**	**Required**	**Type**
token	JWT containing a payment status.	YES	JWT

JWT Header is a constant and does not change:

JWT Header				
Property	**Explanation**	**Required**	**Type**	**Restrictions**
alg	The algorithm used for the signature.	YES	string	The value must be "HS256"
typ	The type of the token.	YES	string	The value must be "JWT"

JWT Payload			
Property	**Explanation**	**Required**	**Type**
transactions	An array of payments and their statuses.	YES	string
transactions. transactionId	The original transaction ID provided by you.	YES	string
transactions.status	Payment initiation amount. Possible status values: • CONFIRMED – the payment has been successfully signed and the funds were received, • SIGNED – the payment has been successfully signed, • FAILED – the payment has failed, • EXPIRED – the payment status is not conclusive (CONFIRMED, SIGNED, FAILED) after a long period of time.	YES	string
iat	The time JWT was issued.	YES	long

Callback request example

HTTP request body


```
{"token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJ0cmFuc2FjdGlvbnMiOlt7InRyYW5zYWN0aW9uSWQiOiJkYzI4MjE2YmU1ZDA0MjI4OGFhOTM1ODRiMjJlYzJhOSIsInN0YXR1cyI6IlNJR05F
RCJ9XSwiaWF0IjoxNTk4MzU3Nzg0MjU5fQ.YD20DZdiX506dHQ9r65SWbhBkv1x7X7wmbwGgdffks8"}
```


Callback JWT

JWT header

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

JWT payload

```
{"transactions":[{"transactionId":"dc28216be5d042288aa93584b22ec2a9","status":"SIGNED"}],"iat":1598357784259}
```

Response Callback

When a callback is sent to you, your system must validate the signature of the JWT to make sure that the callback was sent by us and not someone else. If the signature is correct and you have successfully received the callback, then you should respond with the HTTP status code 200 and the HTTP response body:

HTTP response body

```
{"status":"ok"}
```

In case you consider our callback request invalid, then you should return the HTTP status code 400. The HTTP response body could be added for additional reference like this:

HTTP response body

```
{"status":"failure", "message":"invalid signature"}
```

In case your system is unreachable or returns the HTTP status code 5**, then a callback request will be sent again. You have to respond to our request in 15s.

Prerequisites

Provided by us

The unique contractId that identifies you. This value must be provided in JWT payload during the payment initiation.

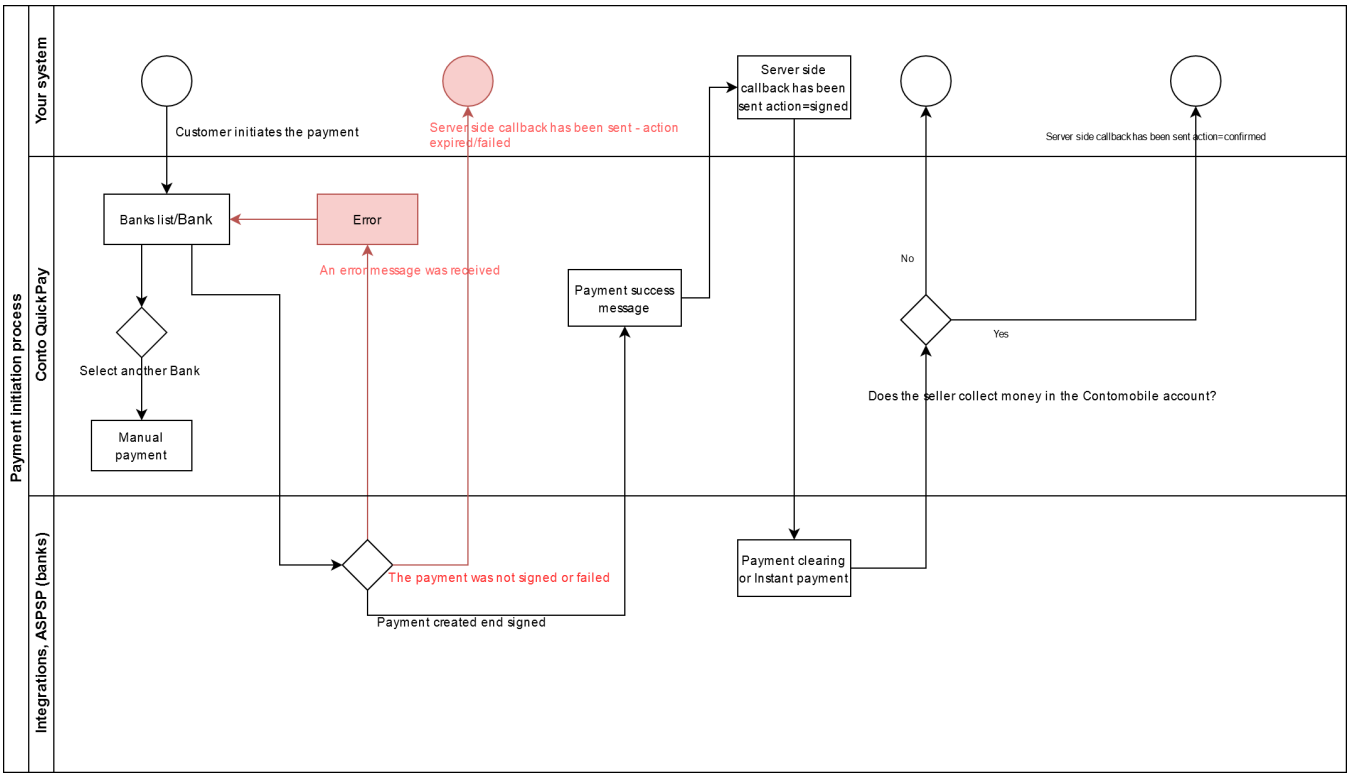
The signing secret is provided by us, it is encoded in base64 and it should be decoded before use. This secret will be used to sign JWT using the HS256 algorithm.

Provided by you

You should provide us with the secure HTTPS endpoint where a callback will be sent, i.e.: https://yourbusiness.com/callback.

Workflow

In order to see the workflow of the whole process, please refer to the below provided diagram with the explanation of each step. We also provide some examples, which contain valid JWT based on a particular secret, therefore they can be used during your development to verify that JWT signatures are correct.



Flow step	Environment	Description
Customers initiates the payment	Your system	The shop initiates the payment that invokes Conto QuickPay widget.
Banks list/Bank	Conto QuickPay	Conto QuickPay widget displays the list of banks to the customer. In it, the customer selects the bank, via which the payment is to be made.
Manual payment	Conto QuickPay	If the customer does not find their online bank in the provided list of banks, selecting Manual payment will open the payment information. The customer will be able to make the payment by entering the previously opened payment orders in their online bank.
Payment created end signed	ASPSP	The payment is created in the payer's bank.
Payment success message	Conto QuickPay	The customer is returned from the bank to Conto QuickPay, where a successful payment message is displayed.
Server side callback has been sent action - signed	Your system	Callback with Signed status is sent to the shop. *If the payment is received instantly, Signed status can be skipped and only Confirmed is sent.*
Payment clearing or Instant payment	ASPSP	Payment clearing or instant payment is made.
Server side callback has been sent action - confirmed	Your system	If the shop collects money on OUR account, callback with Confirmed status is sent to the shop. Otherwise, this step is skipped, i.e. no callback is sent.
The payment was not signed or failed	Your system	In case the user has cancelled the payment or the payment has been rejected by the Bank, we have not received the conclusive status callback.

Example

A 10 euro transaction with the payment purpose set as "MIP test payment" must be made. You are provided with `contractId` that is `71 4462` and the secret that is `Bzg5Js61a0JUXmNT3q2qwoe2YEGuUtxStRILbDEe0zg=`.

Payment initialization

There must be created JWT, which will be used for the Payment initialization. Create JWT with the following parameters:

```
contractId: 714462
amount: 10
paymentPurpose: MIP test payment
transactionId: dc28216be5d042288aa93584b22ec2a9
```

A signed token with the secret `Bzg5Js61a0JUXmNT3q2qwoe2YEGuUtxStRILbDEe0zg=` would look like this:

JWT Token

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJjb250cmFjdElkIjoiNzE0NDYyIiwiYW1vdW50IjoxMCwicGF5bWVudFB1cnBvc2UiOiJNSVAgdGVzdCBwYXltZW50IiwidHJhbnNhY3Rpb25JZCI6ImRjMjgyMTZiZTVkMDQyMjg4YWE5MzU4NGIyMmVjMmE5In0.a-jIP5wuAluCjJRbKQcDcugXwQlS2P2801tOFr8SUwQ
```

And the Payment initialization script would look like this:

Widget initialization

```
            var host = 'https://quickpay.contomobile.com/';
    new QuickPayWidget(host, token, {
        redirectUrlSuccess: 'https://yourshop.com/complete',
                    redirectUrlCancel: 'https://yourshop.com/cancel',
        other: true
    }).initialize();
```

When the payment is initialized, the payment initiation window is displayed.

Payment signed by the user

When the payment has been successfully signed by the user, we redirect the user's browser to your web page at: https://yourshop.com /complete. This web page should display to the user that the made payment was a success.

Signed callback

A callback request will be sent to your server informing you on successful signing of the payment:

HTTP request body

```
{"token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJ0cmFuc2FjdGlvbnMiOlt7InRyYW5zYWN0aW9uSWQiOiJkYzI4MjE2YmU1ZDA0MjI4OGFhOTM1ODRiMjJlYzJhOSIsInN0YXR1cyI6IlNJR05FRCJ9XSwiaWF0IjoxNTk4MzU3Nzg0MjU5fQ.YD20DZdiX506dHQ9r65SWbhBkv1x7X7wmbwGgdffks8"}
```

JWT payload

```
{"transactions":[{"transactionId":"dc28216be5d042288aa93584b22ec2a9","status":"SIGNED"}],"iat":1598357784259}
```

Your server validates JWT and responds to the HTTP call with status 200 and the HTTP response body:

HTTP response body

```
{"status":"ok"}
```

Server side callback

If the account of the beneficiary is a Contomobile account, then you will receive a callback of the confirmed payment.

Request body:

HTTP request body


```
{"token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJ0cmFuc2FjdGlvbnMiOlt7InRyYW5zYWN0aW9uSWQiOiJkYzI4MjE2YmU1ZDA0MjI4OGFhOTM1ODRiMjJlYzJhOSIsInN0YXR1cyI6IkNPTkZJ
Uk1FRCJ9XSwiaWF0IjoxNTk4MzU3Nzg0MjU5fQ.tNd-6uF6DBOOMhWt1jb1LDz74hEM8byQXuQ-8nzSR0Q"}
```


JWT payload


```
{"transactions":[{"transactionId":"dc28216be5d042288aa93584b22ec2a9","status":"CONFIRMED"}],"iat":1598357784259}
```


Your server validates JWT and responds to the HTTP call with status 200 and the HTTP response body:

HTTP response body


```
{"status":"ok"}
```


Payment cancelled by the user

Failed callback

When the payment has been cancelled by the user using the provided navigation buttons, we redirect the user's browser to your web page at: https://yourshop.com/cancel. This web page should display to the user that the payment was cancelled.

A callback request will be sent to your server informing you on the failed payment:

Request body:

HTTP request body


```
{"token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJ0cmFuc2FjdGlvbnMiOlt7InRyYW5zYWN0aW9uSWQiOiJkYzI4MjE2YmU1ZDA0MjI4OGFhOTM1ODRiMjJlYzJhOSIsInN0YXR1cyI6IkZBSUxF
RCJ9XSwiaWF0IjoxNTk4MzU3Nzg0MjU5fQ.RVfkaqiIuaSqkZ3JLRc6q9-bNefztSR0Bua1aPkiQHU"}
```


JWT payload


```
{"transactions":[{"transactionId":"dc28216be5d042288aa93584b22ec2a9","status":"FAILED"}],"iat":1598357784259}
```


Your server validates JWT and responds to the HTTP call with status 200 and the HTTP response body:

HTTP response body


```
{"status":"ok"}
```


Available Banks API

A list of BICs and available banks:

Lithuania:

UAAMLT21XXX - Contomobile UAB
HABALT22XXX - Swedbank AB
AGBLLT2XXXX - Luminor Bank AS Lietuvos filialas
CBVILT2XXXX - AB SEB bankas
INDULT2XXXX - Citadele banka AS Lietuvos filialas
CBSBLT26XXX - AB Šiaulių bankas
VRKULT21XXX - RATO kredito unija
LCKULT22XXX - Lietuvos centrinė kredito unija
MDBALT22XXX - UAB Urbo bankas
RETBLT21XXX - UAB Revolut Bank

Latvia:

HABALV22XXX - Swedbank AS
UNLALV2XXXX - SEB BANKA
RIKOLV2XXXX - Luminor Bank AS Latvian branch
PARXLV22XXX- JSC Citadele BANKA

Estonia:

HABAEE22XXX - Swedbank AS
EEUHEE2XXXX - AS SEB PANK
RIKOEE2XXXX - Luminor Bank AS Estonian branch
PARXEE22XXX- AS Citadele BANKA

Also, a list can be accessed by invoking a service: https://quickpay.contomobile.com/api/setup/country-banks

Latvian and Estonian banks are disabled by default. To enable these banks for a specific contract, contact Support. After activation, they become available as payment options.

> **Optional headers**
>
> When Latvian or Estonian banks are enabled for a contract, the `contractId` header should be included in the request to the bank list service. If the header is not provided, only Lithuanian banks will be returned in the response

An example response could look something like this (it is shortened for explanatory reasons):

HTTP response body

```
GET https://quickpay.contomobile.com/api/setup/country-banks

Headers:
contractId: {yourContractId} - optional

{
    "countryBanks":[
        {
            "countryCode":"LT",
            "banks":[
                {
                    "bic":"UAAMLT21XXX",
                    "active":true
                },
                {
                    "bic":"HABALT22XXX",
                    "active":false
                }
            ],
            "languages":[
                "LIT",
                "ENG"
            ]
        },
        {
            "countryCode":"LV",
            "banks":[
                {
                    "bic":"HABALV22XXX",
                    "active":true
                }
            ],
            "languages":[
                "LAV",
                "ENG"
            ]
        }
    ]
}
```

in this case two banks are available in Lithuania (LT), banks are identified by their BIC: UAAMLT21XXX, HABALT22XXX. A bank identified by UAAMLT21XXX is currently active and can be used right away. Another bank identified by HABALT22XXX is currently not active, this might be because of technical issues or a bank might be updating their system. There are two languages available for this country: LIT and ENG.

Payer notification by email

An email is sent to a payer notifying it if the payment was successful or not. In this context a payer is a buyer, a client of a shop.

JWT additional property

A JWT must contain a `payerNotificationEmail` object property which provides all the parameters required by the function.“address“ property should be encoded using a specific encryption method defined in „Encryption method“. If this function is not required, then this property should not be in the JWT.

JWT payload

```
{
  ...
  "payerNotificationEmail": {
    "address": "zzAvRI64t2V/K/GRtqgY6LmSSb2YcDgggInR4pfm",
    "sendOnSuccess": true,
    "sendOnFailure": true,
    "shopUrl": "https://www.shop.lt",
    "shopName": "shopname",
    "checkoutId": "checkoutIdcheckoutId"
  }...
}
```

All properties are mandatory.

JWT Payload	
Property	**Explanation**
address	An encrypted buyer's email address. A notification will be sent to this address.
sendOnSuccess	Should email be sent if the payment is successful.
sendOnFailure	Should email be sent if the payment is not successful.
shopUrl	Email template value. The website URL of a shop.
shopName	Email template value. The shop name.
checkoutId	Email template value. Shop checkoutId/orderId.

Template values can be found in an example of an email below.

Example scenario 1

When a customer completes a successful payment using "Conto Quickpay" and is redirected to merchant an email is being sent about the payment confirmation.

Contomobile

Užsakymo Nr.: WZAOFNPSM

Mokėjimas sėkmingai inicijuotas!

Informuojame, kad svetainėje parduotuvesadresas.lt
mokėjimas buvo sėkmingai inicijuotas

Mokėjimo suma
0.01 EUR

Pardavėjas:	Jūsų parduotuvės pavadinimas
Užsakymo Nr.:	WZAOFNPSM
Inicijavimo data:	2022 08 30, 09:00:47
Unikalus mokėjimo Nr.	90
Mokėtojo bankas:	Swedbank

Jei turite klausimų dėl prekių pristatymo, paslaugų teikimo ar reikalinga kita, su apmokėtu pirkiniu ar paslauga susijusi informacija, prašome kreiptis į parduotuvesadresas.lt
Rekomenduojame išsaugoti šį laišką, nes tai yra sėkmingai inicijuoto mokėjimo patvirtinimas.

Šį laišką gavote iš Contomobile UAB, kuri yra mokėjimo inicijavimo paslaugos teikėja.

Contomobile UAB
Tuskulėnų 33C-55, 09219 Vilnius, Lietuva
quickpay@contomobile.com

Example scenario 2

When a customer payment using "Conto Quickpay" was not successful due to an error, insufficient funds or expired transaction an email is being sent to retry the payment.

Contomobile

Užsakymo Nr.: YXEDOJDVC

Nesėkmingas mokėjimo inicijavimas!

Informuojame, kad svetainėje parduotuvesadresas.lt
mokėjimas buvo inicijuotas nesėkmingai

Mokėjimo suma
~~**0.01**~~ **EUR**

Pardavėjas:	Jūsų parduotuvės pavadinimas
Užsakymo Nr.:	YXEDOJDVC
Inicijavimo data:	2022 08 30, 09:02:32

Norėdami atlikti mokėjimą, spauskite mygtuką

Bandyti dar kartą

Jei jau esate apmokėję kitais būdais, į šį laišką nekreipkite dėmesio.

Jei turite klausimų dėl prekių pristatymo, paslaugų teikimo ar reikalinga kita, su apmokėtu pirkiniu ar paslauga susijusi informacija, prašome kreiptis į parduotuvesadresas.lt

Šį laišką gavote iš Contomobile UAB, kuri yra mokėjimo inicijavimo paslaugos teikėja.

Contomobile UAB
Tuskulėnų 33C-55, 09219 Vilnius, Lietuva
quickpay@contomobile.com

A full JWT example

JWT Token

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJjb250cmFjdElkIjoiMDAwMDAwMCIsImFtb3VudCI6MC4wMSwicGF5bWVudFB1cnBvc2UiOiJNSVAgdGVzdCBwYXltZW50IiwidHJhbnNhY3Rpb25JZCI6ImMyMjEyOTIyYTViYjQ5NmQ5ZTczYWM2MzU0NmU3NWU0IiwicGF5ZXJOb3RpZmljYXRpb25FbWFpbCI6eyJhZGRyZXNzIjoienpBdlJJNjR0MlYvSy9HUnRxZ1k2TG1TU2IyWWNEZ2dnSW5SNHBmbSIsInNlbmRPblN1Y2Nlc3MiOnRydWUsInNlbmRPbkZhaWx1cmUiOnRydWUsInNob3BVcmwiOiJodHRwczovL3d3dy5zaG9wLmx0Iiwic2hvcE5hbWUiOiJzaG9wbmFtZSIsImNoZWNrb3V0SWQiOiJjaGVja291dElkIiwiY2hlY2tvdXRJZCJ9fQ.l0XMx2WR3HUr1cR2Y4w0jtGiXHaxZv_rj-lAwMEMMc8
```

A `payerNotificationEmail` object "address" property before and after encryption:

HTTP response body

```
...
  "payerNotificationEmail": {
    "address": "email@email.lt",
...
```

HTTP response body

```
...
    "payerNotificationEmail": {
    "address": "zzAvRI64t2V/K/GRtqgY6LmSSb2YcDgggInR4pfm",
...
```

Encryption method

Encryption algorithm must be aes-256-ctr, it is necessary to calculate the sha256 value of secretID to generate iv vector 16 bytes in size. Encryption is required as a raw data. After encrypting the text or data, add the iv vector and the data before it and encode the result in base64. Encrypted data must be the size of encryption algorithm's iv vector's size.

Example in PHP below:

PHP body

```
private function encrypt($data, $secretKey)
{
    $cipherAlgorithm = 'aes-256-ctr';
    $passphrase = openssl_digest($secretKey, 'sha256');
    $ivSize = openssl_cipher_iv_length($cipherAlgorithm);
    $iv = openssl_random_pseudo_bytes($ivSize);
    $cipherOptions = OPENSSL_RAW_DATA;
    $cipherText = openssl_encrypt(
            $data,
            $cipherAlgorithm,
            $passphrase,
            $cipherOptions,
            $iv
    );
    $b64EncodedCipherText = base64_encode($iv.$cipherText);
    return $b64EncodedCipherText;
}Collapse
```

Bulk Payments

Quickpay allows to initiate bulk payments - multiple payments from the same payer to multiple different beneficiaries.

JWT Payload				
Property	**Explanation**	**Required**	**Type**	**Restrictions**
contractId	This property allows us identifying you, and we shall provided it to you at the beginning of the integration.	YES	string	Max length – 140 characters
transactionId	The transaction ID provided by you.	YES	Max36 Text	Max length – 36 characters
exp	The time when JWT expires.	NO	long	
bulkPayments	An array of BulkPayment. See below the definition of the BulkPayment.	YES	array	element count depends on the contract

BulkPayment				
Property	**Explanation**	**Required**	**Type**	**Restrictions**
amount	The payment initiation amount. Note that the payment is only in the EUR currency.	YES	number	Min value – "0.01"
beneficiaryName	Beneficiary name.	YES	string	Max length - 35 characters
beneficiaryAccount	Beneficiary account IBAN.	YES	string	Max length - 34 characters
paymentPurpose	The purpose of the SEPA payment. Values should be unique within the scope of the bulk payment.	YES	string	Max length – 140 characters
transactionId	The transaction ID provided by you. It should be unique for each payment initiation. It allows you to correlate the transaction with its status when the payment status callback is received.	YES	Max36 Text	Max length – 36 characters

An example of a bulk payment JWT payload that has two payments:

JWT payload

```
{
    "contractId": "714462",
    "transactionId": "dc28216be5d042288aa93584b22ec2a0",

    "bulkPayments": [
        {
            "transactionId": "dc28216be5d042288aa93584b22ec2a1",
            "amount": 1.00,
            "beneficiaryName" : "beneficiary1",
            "beneficiaryAccount" : "LT111111111111111111",
            "paymentPurpose": "payment1"
        },
        {
            "transactionId": "dc28216be5d042288aa93584b22ec2a2",
            "amount": 2.00,
            "beneficiaryName" : "beneficiary2",
            "beneficiaryAccount" : "LT222222222222222222",
            "paymentPurpose": "payment2"
        }
    ]
}
```

Callbacks

A bulk payment callback contains a list of payments' statuses.

Payment status	
Status	**Description**
CONFIRMED	The payment has been successfully signed and the funds were received.

SIGNED	The payment has been successfully signed.
FAILED	The payment has been rejected or user-cancelled.
EXPIRED	The payment status is not conclusive (CONFIRMED, SIGNED, FAILED) after a long period of time.

Examples of a callback JWT payload:

JWT payload


```
{
    "bulkPayments": [
        {
            "transactionId": "dc28216be5d042288aa93584b22ec2a0",

            "payments" : [
                {
                    "transactionId": "dc28216be5d042288aa93584b22ec2a1",
                    "status" : "SIGNED"
                },
                {
                    "transactionId": "dc28216be5d042288aa93584b22ec2a2",
                    "status" : "SIGNED"
                }
            ]
        }
    ],
    "iat": 1598357784259
}
```


Only the statuses that have changed will be sent. Later on you may get the following callbacks:

JWT payload


```
{
    "bulkPayments": [
        {
            "transactionId": "dc28216be5d042288aa93584b22ec2a0",

            "payments" : [
                {
                    "transactionId": "dc28216be5d042288aa93584b22ec2a1",
                    "status" : "FAILED"
                }
            ]
        }
    ],
    "iat": 1598357784259
}
```

JWT payload

```
{
    "transactions": [
        {
            "transactionId": "dc28216be5d042288aa93584b22ec2a0",

            "payments" : [
                {
                    "transactionId": "dc28216be5d042288aa93584b22ec2a2",
                    "status" : "CONFIRMED"
                }
            ]
        }
    ],
    "iat": 1598357784259
}
```

Card Payments

Quickpay allows to initiate payments with payment cards.

For card payments a card widget must be initiated. JWT must also contain a cardPayment property with the payer's email.

JWT payload

```
{
...
  "cardPayment": {
    "clientEmail": "UIFyrPVH8iErJe0BJvs3jGdcipWHCktQhtyR6qGHgw=="
  }
}
```

clientEmail property must be encrypted. Encryption method is described here.

The payment is initiated by JavaScript. Therefore, it is necessary to include the following script into your HTML:

JWT payload

```
<script type="text/javascript" src="https://cardpayments.contomobile.com/static/card-widget.js"></script>
```

Pay attention that it is a different script than for the regular payments. This script will allow you to start the payment initiation window by creating a QuickPayCardWidget object.

There are two possible flows when initiating a card payment widget:

- redirect - the payer is redirected to the QuickPay page where the card details form is displayed
- embedded - the card details form is displayed in your page

Redirect flow

Initiate the widget as follows:

JWT payload

```
var host = 'https://cardpayments.contomobile.com/';
new QuickPayCardWidget(host, token, {
    redirectUrlSuccess: 'https://yourshop.com/success_redirect_flow',
    redirectUrlCancel: 'https://yourshop.com/canceled_redirect_flow',
}).initialize();
```

If the flow is successful, then the payer will be redirected to https://yourshop.com/success_redirect_flow and if not, then the payer will be redirected to https://yourshop.com/canceled_redirect_flow.

Embedded flow

Initiate the widget as follows:

JWT payload

```
var host = 'https://cardpayments.contomobile.com/';
new QuickPayCardWidget(host, token, {
    redirectUrlSuccess: null,
    redirectUrlCancel: null,
    embedIn: document.getElementById('cardForm')
})
.onSuccess(function () {
    window.location = 'https://yourshop.com/success_redirect_flow';
})
.onCancel(function () {
    window.location = 'https://yourshop.com/canceled_redirect_flow';
}).initialize();
```

In this case the card details form will be displayed on your web page. The form will be embedded in the HTML element with ID cardForm. JavaScript functions must be defined for success and failure cases.

For the embedded flow to work you might also need to adjust Content Security Policy (CSP) of your web page to allow the iframe from the source:

JWT payload

```
<iframe src="https://cardpayments.contomobile.com/initialize?token=...">
```

Callback

Callbacks work the same as for the regular payments. It is not enough that the user is redirected to your success or failure pages, the true outcome of the payment is determined by the callback that we send to you. A system callback request is sent to your system informing you about the status of the payment. No system callback will be sent if the user closes the widget or the browser before he /she is requested to authorize the payment.

Payment status CONFIRMED is returned, if the user has successfully completed the payment; or status FAILED is returned, if the payment was not completed or rejected.

Status EXPIRED may be returned, if the payment status is not conclusive (FAILED, SIGNED, CONFIRMED) for a long period of time. This can happen when the user closes the browser; in such a case, the bank of the user might not change the payment status from pending to a more conclusive.

Transaction status

Status	Description
CONFIRMED	The payment has been successfully signed and the funds were received.
FAILED	The payment has been rejected.
PENDING	The payment status (CONFIRMED, FAILED) has not been received yet.

JWT Example

For example secret key:

JWT Token

```
Bzg5Js61a0JUXmNT3q2qwoe2YEGuUtxStRILbDEe0zg=
```

The payer email is email@email.com, then the JWT would look like this:

JWT payload

```
{
  "alg": "HS256",
  "typ": "JWT"
}
{
  "contractId": "714462",
  "amount": 10,
  "paymentPurpose": "MIP test payment",
  "transactionId": "dc28216be5d042288aa93584b22ec2a9",
  "payerNotificationEmail": null,
  "cardPayment": {
    "clientEmail": "UIFyrPVH8iErJe0BJvs3jGdcipWHCktQhtyR6qGHgw=="

  }
```

Signed and encoded:

JWT Token

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJjb250cmFjdElkIjoiNzE0NDYyIiwiYW1vdW50IjoxMCwicGF5bWVudFB1cnBvc2UiOiJNSVAgdGVzdCBwYXltZW50IiwidHJhbnNhY3Rpb25JZCI6ImRjMjgyMTZiZTVkMDQyMjg4YWE5MzU4NGIyMmVjMmE5IiwicGF5ZXJOb3RpZmljYXRpb25FbWFpbCI6bnVsbCwiY2FyZFBheW1lbnQiOnsiY2xpZW50RW1haWwiOiJVSUZ5clBWSDhpRXJKZTBCSnZzM2pHZGNpcFdIQ2t0UWh0eVI2cUdIZ3c9PSJ9fQ.YiNjx8IH6IgXNV72b4UPX5-
ZnJof7AruT_OnjpVl3Do
```

Use https://jwt.io/ for experimentation, but do not use the production secret there.

Sandbox

Payment initiation Sandbox example

HTML snippet

```
<script type="text/javascript" src="https://pis-sandbox.contomobiledev.com/static/widget.js"></script>
```

Widget sandbox initialization example

Widget initialization

```
            var host = 'https://pis-sandbox.contomobiledev.com';
    new QuickPayWidget(host, token, {
        redirectUrlSuccess: 'https://yourshop.com/success_redirect_flow',
                    redirectUrlCancel: 'https://yourshop.com/canceled_redirect_flow',
        defaultLanguage: 'LIT',
        defaultCountry: 'LT',
        creditor: 'UAAMLT21XXX',
        other: true,
        webview: false,
    }).onSuccess(function () {
        // add your logic on success
    }).onCancel(function () {
        // add your logic on cancel
    }).initialize();
```

Card payment initiation Sandbox example

HTML snippet

```
<script type="text/javascript" src="https://cardpayments-sandbox.contomobiledev.com/static/card-widget.js"><
/script>
```

Card payment is rejected in the sandbox environment if the payment 'amount' is 0.02. Otherwise, the payment is successful.

Redirect flow

Redirect widget initialization example:

JWT payload

```
var host = 'https://cardpayments-sandbox.contomobiledev.com/';
new QuickPayCardWidget(host, token, {
    redirectUrlSuccess: 'https://yourshop.com/success_redirect_flow',
    redirectUrlCancel: 'https://yourshop.com/canceled_redirect_flow',
}).initialize();
```

Embedded flow

Embedded widget initialization example:

JWT payload

```
var host = 'https://cardpayments-sandbox.contomobiledev.com/';
new QuickPayCardWidget(host, token, {
    redirectUrlSuccess: null,
    redirectUrlCancel: null,
    embedIn: document.getElementById('cardForm')
})
.onSuccess(function () {
    window.location = 'https://yourshop.com/success_redirect_flow';
})
.onCancel(function () {
    window.location = 'https://yourshop.com/canceled_redirect_flow';
}).initialize();
```

Periodic Payments

Quickpay allows to initiate periodic/recurring payments over the specified schedule (timeframe).

JWT Payload

Property	Explanation	Required	Type	Restrictions
contractId	This property allows us identifying you, and we shall provided it to you at the beginning of the integration.	YES	string	Max length – 140 characters
amount	Payment amount.	YES	number	Min value – “0.01”
paymentPurpose	SEPA payment purpose.	YES	string	Max length – 140 characters
transactionId	The transaction ID provided by you.	YES	Max36 Text	Max length – 36 characters
exp	The time when JWT expires.	NO	long	Unix Epoch Time
serviceType	The type of service. In case of one-time paymenPeriodic Paymentst not required. In case of periodic payment required.	YES	string	Constant - "PERIODIC"
periodicPayment.startDate	The start date of the periodic payment.	YES	string	ISODate - cannot be earlier than tomorrow (for some banks 3 days in the future)

periodicPayment.endDate	The end date of periodic payment.	YES	string	ISODate - cannot be earlier than "startDate"
periodicPayment.frequency	The frequency of periodic payment.	YES	string	Possible values - "Daily", "Weekly", "Monthly", "Annual". Values can depend on the bank

An example of JWT payload for periodic payment:

```
{
   "contractId":"713106",
   "amount":0.01,
   "paymentPurpose":"MIP test payment",
   "transactionId":"3b7224056250404188 37bfb3a4e68b57",
   "payerNotificationEmail":null,
   "cardPayment":null,
   "serviceType":"PERIODIC",
   "periodicPayment":{
      "startDate":"2025-04-03",
      "endDate":"2025-04-09",
      "frequency":"Daily"
   }
}
```

Callbacks

The callback returned in case of periodic payment specifies the status of periodic payment **agreement** itself and **not** of particular scheduled payments made according to the periodic payment agreement.

Periodic payment agreement status	
Status	**Description**
ACTIVE	The status of the periodic payment agreement, as verified with the bank, is successfully signed and active/valid.
CANCELLED	The status of the periodic payment agreement, as verified with the bank, is unsigned (e.g., due to an error), cancelled, or otherwise inactive/invalid.
UNRESOLVED	The status of the periodic payment agreement is unknown – the signing session has expired.

Cancellation of periodic payment agreement

The periodic payment agreement can only be cancelled on the bank itself by the shop customer, i. e. Quickpay does not provide the possibility to cancel periodic payment agreement via Quickpay integration.

A few examples of JWT payload for callbacks:

In case of periodic payment agreement being ACTIVE:

```
{
  "transactions": [
    {
      "transactionId": "dDQK6SMkhv6ik",
      "status": "ACTIVE"
    }
  ],
  "iat": 1749207203
}
```

In case of periodic payment agreement being CANCELLED:

```
{
  "transactions": [
    {
      "transactionId": "dDQK6SMkhv6ik",
      "status": "CANCELLED"
    }
  ],
  "iat": 1749208989
}
```

Payment Refund

Quickpay supports full or partial payment refunds.

To initiate a payment refund an HTTP request with a signed JWT must be made to Quickpay:

```
POST https://quickpay.contomobile.com/api/payment/refund

{JWT}
```

an HTTP status 201 is returned if the request was successful.

JWT Payload

Property	Explanation	Required	Type	Restrictions
contractId	This property allows us identifying you, and we shall provided it to you at the beginning of the integration.	YES	string	Max length – 140 characters
amount	Amount to be refunded. Partial refund is supported.	YES	number	Min value – "0.01" Max value - original payment amount or what's left of it
paymentPurpose	SEPA payment purpose.	YES	string	Max length – 140 characters
transactionId	A new transaction ID for the refund itself.	YES	Max36 Text	Max length – 36 characters
originalTransactionId	The original transaction ID of the payment that is refunded.	YES	Max36 Text	Max length – 36 characters
creditorName	Refund beneficiary name.	YES	string	

Callback

A callback with a status CONFIRMED and transactionId is sent to the shop when the money is subtracted from the shop's account.

Example

Original transaction ID is dc28216be5d042288aa93584b22ec2a9 and a shop wants to refund 1.00 EUR to the buyer. Then JWT payload could look like this:

JWT payload

```
{
  "contractId": "714462",
  "amount": 1,
  "paymentPurpose": "MIP test payment",
  "transactionId": "dc28216be5d042288aa93584b22ec111",
  "originalTransactionId": "dc28216be5d042288aa93584b22ec2a9",
  "creditorName": "Buyer's Name"
}
```

and an HTTP request would look like this:

HTTP request

```
POST https://quickpay.contomobile.com/api/payment/refund

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJjb250cmFjdElkIjoiNzE0NDYyIiwiYW1vdW50IjoxLjAwLCJwYXltZW50UHVycG9zZSI6Ik1JUCB0ZXN0IHBheW1lbnQiLCJ0cmFuc2FjdGlvbklkIjoiZGMyODIxNmJlNWQwNDIyODhhYTkzNTg0YjIyZWMxMTEiLCJvcmlnaW5hbFRyYW5zYWN0aW9uSWQiOiJkYzI4MjE2YmU1ZDA0MjI4OGFhOTM1ODRiMjJlYzJhOSIsImNyZWRpdG9yTmFtZSI6IkJ1eWVyJ3MgTmFtZSJ9.9lYaOHt54SG7oGHojmwORsOGA_l8FaLz4c-Se6P3vY4
```

after a while a callback with a status CONFIRMED is sent to the shop.

Callback JWT payload

```
{"transactions":[{"transactionId":"dc28216be5d042288aa93584b22ec111","status":"CONFIRMED"}],"iat":1598357784259}
```