

## OP Online Payment

These are the steps for processing a Op Online transaction with the API:

1. Generate a unique `merchant_reference` that will be linked to this transaction. This reference needs to be used to link the order of the customer with the transaction id generated by Verifone.
2. Perform the create transaction API call and set the `return_url`, `cancel_url`, `reject_url` to include the `merchant_reference` so you can display the transaction result when they return to your server. The `*_url` can include the reference as part of the url ( `/order/1234` ) or as a query parameter ( `/order?merchant_reference=1234` ).
3. Redirect the shopper to the `payment_url` provided by Op Online in the response of the API call. When the customer has completed the payment on their telephone, they are returned to the `*_url` you provided in the request body.
4. When the customer visits the `return_url`, `cancel_url`, `reject_url` you initially provided, there are two options for retrieving the transaction status:
  - Use the [notification](#) functionality and wait until the webhook is received that the transaction has been completed ("eventType":"TxnSaleApproved")
  - Query the `GET` transaction endpoint using the id returned in the API call from step 3

## Required fields

Parameters	Description
<code>payment_provider_contract</code>	In the <a href="#">Payment Provider Contracts</a> section in Verifone Central, set the <i>Payment Type</i> to <i>Plaid</i> , select your contract and copy the Payment Provider Contract ID. Note this value is different in Sandbox and in Production.
<code>bank_id</code>	OP bank ID for the bank to be used.
<code>amount</code>	Amount of the transaction.
<code>customer</code>	<a href="#">Customer ID</a>
<code>merchant_reference</code>	Unique UUID you generate and can link the transaction to when the customer returns.
<code>currency_code</code>	Accepted values are EUR.
<code>return_url</code>	Generate a unique link and include the <code>merchant_reference</code> either as a query parameter or as part of the URL ( <a href="https://merchant.com/order/1234">https://merchant.com/order/1234</a> or <a href="https://merchant.com/order?merchant_reference=1234">https://merchant.com/order?merchant_reference=1234</a> ). Endpoint where customer's browser is redirected to at the end of the browser flow when the payment was successful.
<code>cancel_url</code>	Endpoint where customer's browser is redirected to at the end of the browser flow when the payment was cancelled.
<code>reject_url</code>	Endpoint where customer's browser is redirected to at the end of the browser flow when the payment failed.

## Generate a unique merchant\_reference

The `merchant_reference` needs to be unique to identify the shopper when they are redirected to your server by Op Online. In the next step, you will be creating a transaction through the API, the transaction will return an id that needs to be stored safely with the reference. When the shopper returns, you can use this reference to confirm either through the webhook or through the `GET` transaction API call if the shopper has successfully completed the transaction.

## Create Transaction API call

`POST` the following example to the Op Online Payment endpoint: `/oidc/api/v2/transactions/op-online-payment`.

Unsure what URL you should be using? The full list can be found [here](#).

### Request

#### Headers

URL	<a href="#">List of available servers</a>
Endpoint	<code>/oidc/api/v2/transactions/op-online-payment</code>
x-vfi-api-idempotencyKey	Unique UUID to identify the transaction
Authorization	Basic {{your encoded user id and api secret}}
Content-Type	application/json

#### Body

```
{
  "payment_provider_contract": "replace with your own payment_provider_contract",
  "bank_id": "replace with your bank_id",
  "amount": 100,
  "customer": "replace with your own customer object",
  "merchant_reference": "generate a unique merchant_reference",
  "currency_code": "EUR"
  "return_url": "https://yourwebsite.com/order/replace with your url for successful payments",
  "cancel_url": "https://yourwebsite.com/order/replace with your url for cancelled payments",
  "reject_url": "https://yourwebsite.com/order/replace with your url for rejected payments",
}
```

### Response

The `return_url` in the request is the site where the shopper is sent **after** completing the transaction. The `payment_url` that is returned in the response is the url used by **Op Online Payment** the shopper must visit to complete the payment.

From the response, the id of the transaction should now be stored along with the `merchant_reference` from the first step. When the shopper returns to your environment, you can confirm the `merchant_reference` by cross-referencing it with the transaction status using the id.

You should now use the `payment_url` from the **response** to redirect the shopper to Op Online Payment website.

```
{
  "amount": 100,
  "status": "INITIATED",
  "id": "df945024-114e-4885-a97e-9bec2a0f33ec",
  "payment_url": "https://api.smn-sandbox.aws.op-
palvelut.fi/customer/payment/multibank/direct/BankoEnablado/startPaymentConfirmation?paymentOperationId=4778b7e1-
3b77-4a17-8611-74b5a5cf83e7be4097be-b03c-4959-8389-
f74e73b2e966&paymentOperationValidation=OmCGqOXCvVmJdOx4Ak5c3NCWjJYEEntZcSHqr9gpEyg%3D",
  "processor": "OP_ONLINE_PAYMENT",
  "payment_product": "OP_ONLINE_PAYMENT"
}
```

## Redirect the shopper

Redirect the shopper to the `payment_url` from the previous step. The shopper will now complete the Op Online payment. After completion or cancellation, the shopper will be sent back to the `payment_url`.

## Retrieving the transaction status

When the customer is returned to your environment, you have two options to confirm the status of the transaction:

1. Confirm the transaction through the [notification service](#) with webhooks/emails
2. Query the `GET` transaction endpoint `/oidc/api/v2/transaction/{id}` using the id from the response in step 2

### Retrieving the transaction status using the notification service

Following the steps in the [notification service documentation](#), set up a webhook to be sent for the event type `TxnSaleApproved` and `TxnSaleDeclined` for your organization. When Verifone receives a notification that the transaction has been completed or declined, the webhook will be sent. Here is an example of a webhook:

```
{
  "eventId": "1",
  "eventDateTime": "2022-03-23T11:07:28Z",
  "recordId": "transaction id",
  "eventType": "TxnSaleApproved"
}
```

The `recordId` field will contain the transaction ID, and the `eventType` field can be parsed to view the outcome of the transaction. For the response in step 2 of this guide, you stored the combination of the `merchant_reference` and transaction id.

### Retrieving the transaction status by querying `/oidc/api/v2/transaction/{id}`

Alternatively, the direct API can be used to query the transaction status. After the shopper returns to your site, you can do a `GET` request to search for the status of the transaction:

#### Request

Headers

URL	<a href="#">List of available servers</a>
Endpoint	/oidc/api/v2/transaction/{id}
Authorization	Basic {{your encoded user id and api secret}}
Content-Type	application/json

### Response

The response will have the status field which can be used to determine the outcome of the transaction.

```
{
  "id": "df945024-114e-4885-a97e-9bec2a0f33ec",
  "amount": "158.56",
  "currency_code": "EUR",
  "created_at": "2023-08-21T14:45:05.796Z",
  "shipping_information": {},
  "status": "SALE SETTLED",
  "processor_reference": "4778b7e1-3b77-4a17-8611-74b5a5cf83e7",
  "cvv_present": false,
  "shopper_interaction": "ecommerce",
  "merchant_id": "330303",
  "payment_summary": {
    "captured_amount": "158.56"
  },
  "transaction_status": "SETTLED",
  "transaction_type": "SALE"
}
```