OP Online Payment

Overview

The OP Online API integration guide walks you through the steps that you need to complete to accept transactions using OP Online advanced payment method.

Etymology:

- **Client** - is a computer or software program that requests services or resources from a server over a network.
- A **server** is a computer or software program that provides services or resources to other computers, known as clients, over a network. Servers are designed to handle requests from clients and fulfill those requests by providing data, processing tasks, or performing computations.

Prerequisites

- This guide assumes a basic understanding of HTML and JavaScript.
- This integration also requires the ability to make server-side REST API calls. **API calls should not be made from the client-side**.

Before you get started

Starting from scratch? Follow these steps to get started from the beginning.

- Have a Verifone Central account in either our Sandbox or Production environment with API Access.
- Don't have a Verifone Central Account already? Contact your regional sales team to get started.
- Have access to a Verifone Central account that has the Merchant Cashier or Merchant Supervisor user role.
- Forgotten your password? Click the portal link for your region and reset your password.
- Generate a Secure Card Capture Key on your user's organization.
- (Optional) Activate *Advanced Payment Methods* to provide more channels to customers:
    - Check which APM's your acquirer supports
    - Activate Apple Pay for Checkout
    - Activate Google Pay for Checkout
    - Activate your Paypal account

Setup API access

Setting up Authentication with BasicAuth

BasicAuth with Verifone requires a username and password. Both your **username** (userID) and **password** (API key), can be collected from Verifone Central.

Collecting your Verifone credentials (environment variables)

- Select an environment (Sandbox or Production)
    - Make sure you set you set the base of the URL (environment) up to be configurable for when you are ready to proceed to production.
- Collect your Organization ID
    - Your organization(s) provide a structure to access different sales channels across your businesses and stores. The Organization ID is referenced to target a particular merchant site for the payment.
- Collect your Payment Provider Contract ID
    - The Payment Provider Contract (PPC) represents the merchants contract with their acquiring partner, including key details such as the Merchant ID and supported currencies.

OP Online payment

Step 1: 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. The `merchant_reference` can be any string associated to your CRM or internal system, or a random unique string associated to this transaction.

After the transaction is created, you can later use the `merchant_reference` to retrieve information related to the created transaction.

Step 2: Perform the create transaction API call

API Reference: https://verifone.cloud/api-catalog/verifone-ecommerce-api#tag/Ecom-Payments/operation/saleTransaction

Make a POST request to create the merchant initiate transaction. The response is the payment URL.

Request Method: POST

URL: {environment}/oidc/api/v2/transactions/op-online-payment.

Headers:

x-vfi-api-idempotencyKey - unique UUID to identify the transaction

> **Important**: To check which parameters are required check our eCommerce API documentation for OP Online Payment.

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** successfully completing the transaction.

The `cancel_url` that is returned in the response and is the URL where the shopper is redirected after cancelling a payment session.

The `reject_url` that is returned in the response is the URL where the shopper is redirected if the payment is rejected from a technical or other reason.

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"
}
```


status - indicates the outcome of the transaction.
id - the ID of the transaction. Details of the transaction along with the result can be queried using the Get Transaction by ID API call.
payment_url - the URL which can be accessed from any browser. It should be used to redirect the shopper.

Step 3: 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`.

Step 4: Retrieve 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 using Get Transaction by ID API Call

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:

API Reference: https://verifone.cloud/api-catalog/verifone-ecommerce-api#tag/Transaction/operation/readTransaction

Request Method: GET

Path parameters: id – string (**example**: 76944d4b-89e6-48d2-ac04-675383c3eedf)

URL: {environment}/oidc/api/v2/transaction/{id}

Response:


```
{
"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"
}
```


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