**EMS Online – API integration manual**

**Version: 0.1 (draft)**

**Step 1 - get api key**

Get (test) API key via merchant portal

**Step 2 Setup authentication**

An API client must authenticate itself for every request to the API. The client can do this by using HTTP Basic Authentication. The username is the API key as generated by EMS Online, the password remains empty.

An example of an Authorization header:

Authorization: Basic aHVudGVyMjo=

The value after “Basic” can be obtained by encoding the API key (followed by a colon) using the Base64 algorithm. The API key hunter2followed by : will result in the string aHVudGVyMjo=.

Most HTTP clients support Basic Authentication out of the box.

The API URL to be used is: [https://api.online.emspay.eu](https://api.online.emspay.eu/)

**Step 3 - Construct HPP order**

You can also use our hosted payment page instead of a custom implementation or plugin. When using this functionality as part of your checkout process, we recommend to use a regular HTTP redirect to automatically forward the consumer to the hosted payment page.

You can send an order request without any transactions (payment method).

It is optional to set an expiration\_period, which will expire the payment page after a fixed [duration](http://en.wikipedia.org/wiki/ISO_8601#Durations). In the example below, the expiration period for the payment page URL is set as 30 days.

The payment page URL is returned in order\_url field of the response, which the consumer can follow to complete the payment.

Example order request:

POST /v1/orders/ HTTP/1.1

Authorization: Basic aHVudGVyMjo=

Content-Type: application/json

{

"currency": "EUR",

"amount": 995,

"description": "Example description",

"expiration\_period":"P30D",

"customer": {

"locale": "en-GB"

}

}

On the hosted payment page, all payment methods available to the consumer are shown. These are the payment methods that have been approved and are enabled for the webshop.

Customer data can contain a preferred locale, possible locales are: nl, de, nl-BE, fr-BE, en-GB. This locale will be used to determine the language of the hosted payment page.

**Step 4 - Interpreted response**

Response:

{

"amount": 995,

"client": {

"user\_agent": "Testing API"

},

"created": "2016-07-04T11:41:57.121017+00:00",

"currency": "EUR",

"description": "Example description",

"id": "1c969951-f5f1-4290-ae41-6177961fb3cb",

"merchant\_id": "7131b462-1b7d-489f-aba9-de2f0eadc9dc",

"modified": "2016-07-04T11:41:57.183822+00:00",

"order\_url": "https://api.online.emspay.eu/pay/1c969951-f5f1-4290-ae41-6177961fb3cb/",

"project\_id": "1ef558ed-d77d-470d-b43b-c0f4a131bcef",

"status": "new"

}

The resulting payment page URL links directly to the payment page for this order:

<https://api.online.emspay.eu/pay/a8c1b7b0-9c73-4bd8-94f4-37c3070e78c7/>

**Step 5 - setup webhook**

In some cases, the consumer does not return to the webshop (in the case of bank transfer, the payment will always be decoupled), and another way of checking the current status of the order needs to be used. It is possible to periodically retrieve the order and check whether the status was changed, but this is inefficient and cumbersome.

To solve this, webhooks can be used. A webhook is a callback (HTTP POST) to a merchant-supplied URL. The URL is configurable per project in the webhook\_url field. The order ID and project ID are available in the body of the request. EMS Online makes the request whenever the status of an order changes.

Example webhook from EMS Online to <http://www.example.com/webhook>:

POST /webhook HTTP/1.1

Host: www.example.com

Content-Type: application/json

{

"event": "status\_changed",

"order\_id": "4c6afd74-a840-4aab-b411-1e6e0636d341",

"project\_id": "b5f39273-44e7-4385-8e08-44612ef3e117"

}

Webhooks can be configured on project level, but can be overwritten for a specific order by supplying a webhook url when creating the order. For this specific order the specific webhook url will be used. This is part of the order schema:

{

"$schema": "http://json-schema.org/draft-04/schema#",

"title": "Order",

"type": "object",

"properties": {

"id": {

"description": "Order identifier",

"type": "string",

"readOnly": true

},

"webhook\_url": {

"description": "Used for transaction updates",

"type": "string",

"format": "uri"

}

}

}